



FAKULTI SAINS KOMPUTER

&

TEKNOLOGI MAKLUMAT

UNIVERSITI MALAYA,

KUALA LUMPUR.

Perpustakaan SKTM

EMAIL VIRUS SCANNER V1.1

(EMVS V1.1)

Disediakan :

NUR FIRDAUS GHAZALI

WEK000161

Penyelia :

PUAN FAZIDAH OTHMAN

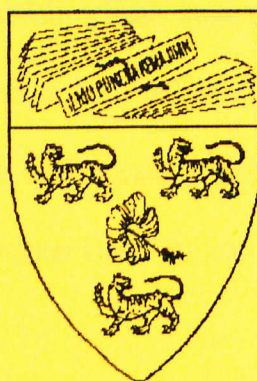
Moderator:

ENCIK NOR BADRUL ANUAR JUMA'AT

Perpustakaan Universiti Malaya



A511275711



ABSTRAK

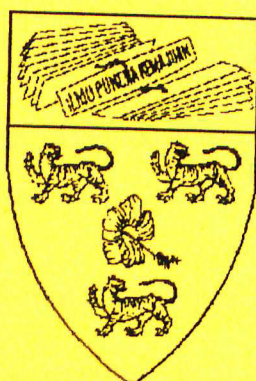
ABSTRAK

Email Virus Scanner adalah sebuah system pengimbas mudah alih dan senang digunakan. Sistem ini bersaiz kecil jadi ia mudah dibawa ke mana-mana dan tidak perlu dipasang di dalam sistem pengoperasian. Sasaran utama sistem ini adalah pelajar Fakulti Sains Komputer Dan Teknologi Maklumat, Universiti Malaya.

Pembangunan sistem ini adalah untuk mengurangkan kebergantungan terhadap sistem yang telah sedia ada yang terdapat pada sesebuah komputer. Sistem ini akan dapat membantu mengurangkan ancaman serangan virus pada komputer pengguna.

Email Virus Scanner V1.1 (EMVS V1.1) adalah peningkatan kepada Email Virus Scanner V1.0 (EMVS V1.0) yang telah sedia ada. Dalam EMVS V1.1 beberapa perubahan telah dilakukan supaya ia lebih cekap dan berkualiti.

Diharap EMVS V1.1 ini akan dapat memperbaiki kelemahan yang terdapat pada EMVS V1.0 dan seterusnya akan dapat digunakan secara meluas terutamanya di FSKTM, UM.



PENGHARGAAN

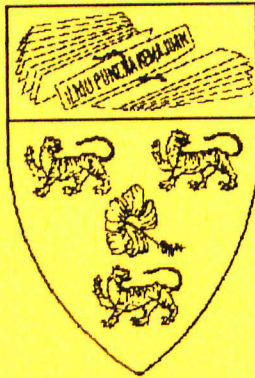
PENGHARGAAN

Syukur saya ke hadrat Ilahi kerana dengan limpah rahmatNya dapat saya menyiapkan Laporan Ilmiah Tahap Akhir I & II ini.

Pertama sekali saya ingin merakamkan jutaan terima kasih kepada penyelia saya, Puan Fazidah Othman yang telah banyak memberi pandangan, komen serta tunjuk ajar sepanjang saya menyiapkan projek ini. Juga kepada moderator saya iaitu Encik Nor Badrul Anuar Juma'at yang telah banyak memberi saranan dan komen yang berguna kepada saya.

Terima kasih yang tidak terhingga kepada ahli keluarga saya, MJ dan rakan-rakan yang banyak memberi semangat, bantuan dan tujuk ajar kepada saya untuk menyempurnakan projek ini.

Terima Kasih.



KANDUNGAN

University of Malaya

KANDUNGAN

ABSTRAK	ii
---------	----

PENGHARGAAN	iii
-------------	-----

BAB 1 PENGENALAN

1.1	Pengenalan Projek	1
1.2	Motivasi Projek	3
1.3	Objektif system	3
1.4	Skop Projek	4
1.5	Had Sistem	5
1.6	Kepentingan Sistem	6
1.7	Keperluan Sistem	6
1.8	Hasil Yang Dijangkakan	7
1.9	Penjadualan Projek	8

BAB 2 KAJIAN LITERASI

2.1	Pengenalan	12
2.2	Pengenalan Kepada Email Virus Scanner	13
2.3	Kaedah Kajian	24
2.4	Analisis Sistem Sedia Ada	28
2.5	Perbandingan Sistem	30

BAB 3 METODOLOGI

3.1	Metodologi Sistem	32
3.2	Jenis Metodologi	34
3.3	Pilihan Metodologi	34
3.4	Struktur Model	36

BAB 4 ANALISIS SISTEM

4.0	Spesifikasi Keperluan Sistem	41
4.1	Keperluan Fungsian	41
4.2	Keperluan Bukan Fungsian	43
4.3	Keperluan Perisian	46
4.4	Keperluan Sistem Pengoperasian	49

BAB 5 STRUKTUR SISTEM

5.1	Rekabentuk Sistem	53
5.2	Rekabentuk Antaramuka Grafik	53
5.3	Rekabentuk Pangkalan Data	55
5.4	Rekabentuk EMVS	57
5.5	Carta Aliran Sistem	61
5.6	Rekabentuk Antaramuka pengguna	61
5.7	Antaramuka Pengguna Yang Dicapangkan	62
5.8	Gambarajah Aliran Data	64

BAB 6 PERLAKSANAAN SISTEM

6.1	Pengenalan	68
6.2	Proses – Proses Yang Terlibat	69
6.3	Pengaturcaraan	70

BAB 7 PENGUJIAN SISTEM

7.1	Pengenalan	72
7.2	Jenis – Jenis Ralat	73
7.3	Jenis –Jenis Pengujian	75

BAB 8 PERBINCANGAN

8.1	Pengenalan	80
8.2	Kekuatan Sistem	80
8.3	Kekangan Sistem	82
8.4	Perancangan Masa Depan	82
8.5	Masalah Dan Penyelesaian	83
8.6	Cadangan	84
8.7	Kesimpulan	85

APENDIDIKS

BIBLIOGRAFI

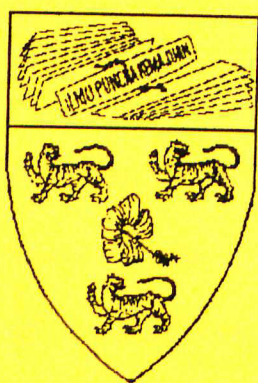
USER MANUAL

SENARAI JADUAL

Jadual 1.1	Garis Masa Pembangunan Sistem	11
-------------------	--------------------------------------	-----------

SENARAI RAJAH

Rajah 3.1	Model Air Terjun Dengan Prototaip	37
Rajah 3.2	Prototaip ‘Evolutionary’	39
Rajah 5.1	Carta Hierarki EMVS	58
Rajah 5.2	Sub Modul Reports	59
Rajah 5.3	Sub Modul Options	60
Rajah 5.4	Carta Aliran Bagi EMVS	61
Rajah 5.5	Rekabentuk Antaramuka Log In	62
Rajah 5.6	Rekabentuk Antaramuka Scan	63
Rajah 5.5	Rekabentuk Antaramuka Reports	63
Rajah 7.1	Skema Ujian Modul	76
Rajah 7.2	Skema Ujian Sistem Teknik Bawah – Atas	77



BAB I PENGANTARAN

1 PENGENALAN

1.1 Pengenalan Projek

Saya telah memilih projek **Email Virus Scanner V1.1 (EMVS V1.1)** sebagai tugas yang akan dilaksanakan semasa menjalani Latihan Ilmiah Tahap Akhir (WXES 3181/3182). System ini adalah peningkatan kepada system Email Virus Scanner V1.0 yang telah sedia ada. Beberapa perubahan akan dilakukan supaya system ini menjadi lebih cekap dan bermutu.

Projek ini adalah merupakan suatu sistem pengimbas virus flexibel mudah alih. Sistem ini dikatakan mudah alih kerana ianya bersaiz kecil iaitu hanya memerlukan satu Compact Disc (CD) sahaja dan dapat di bawa ke mana-mana serta mudah digunakan. Pengguna hanya perlu mengklik satu ikon sahaja dan sistem ini berupaya untuk mengimbas beberapa jenis virus yang berbahaya. Virus-virus yang berupaya dikesan oleh sistem ini adalah virus-virus yang biasanya diterima daripada emel dan disebabkan itu juga ia dinamakan **Email Virus Scanner (EMVS)**. Sistem ini tidak seperti perisian antivirus lain yang perlu dipasang dahulu di dalam sistem pengoperasian sebelum boleh digunakan. Sistem yang akan mengimbas virus emel pada setiap komputer ini diharap akan digunakan secara meluas terutamanya oleh pelajar Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Malaya, Kuala Lumpur.

Fakulti Sains Komputer dan Teknologi Maklumat ini telah ditubuhkan pada tahun 1965 dan dahulunya dikenali sebagai Pusat Komputer Universiti. Pusat ini juga

dahulunya telah menjadikan universiti ini sebagai perintis penggunaan komputer di Malaysia.

Memandangkan perkembangan dunia Internet kini, adalah penting bagi seseorang pengguna mempunyai sistem yang boleh mengimbas virus emel yang mudah alih. Ini adalah penting kerana ancaman virus yang ada sekarang telah banyak menyebabkan kerugian yang besar khususnya kepada organisasi mahupun orang perseorangan.

Sistem yang dibangunkan menggunakan perisian Visual Basic 6.0 ini akan mengimbas virus emel yang terdapat pada komputer pengguna tanpa bergantung sepenuhnya kepada antivirus sedia ada. Sebarang virus yang dikesan akan terus dihapuskan dan laporan berkaitan aktiviti pengimbasan akan dilampirkan bersama. Sistem ini akan direka dan dibangunkan dengan berkonsepkan ramah pengguna. Ini penting agar pengguna dapat menggunakan sistem ini dengan mudah. Bahasa sistem dan antaramuka yang digunakan juga adalah jelas, terang serta mudah difahami. Email Virus Scanner (EMVS) ini akan digunakan terutamanya di Fakulti Sains Komputer dan Teknologi Maklumat Universiti Malaya.

Tujuan menggunakan sistem ini adalah untuk memastikan tahap keselamatan yang optimum terhadap sebarang ancaman virus yang mungkin boleh mengganggu sistem atau menyebabkan kerosakan pada perisian mahupun perkakasan komputer. Selain daripada itu, ia juga adalah untuk mempermudah pengguna terutamanya mereka yang gemar menggunakan perkhidmatan komputer samaada di makmal fakulti, pusat-pusat Internet mahupun di tempat-tempat lain kerana mereka boleh

sistem ini ke mana-mana sahaja dan hanya dengan mengklik satu ikon sahaja sistem ini sudah mampu untuk mengimbas beberapa jenis virus yang merbahaya.

1.2 Motivasi Projek

Kebanyakan komputer yang terdapat pada mana-mana universiti mahupun organisasi biasanya hanya bergantung kepada satu perisian antivirus sahaja. Kepada yang peka terhadap persekitaran ancaman virus, mereka mungkin sentiasa mengemaskini perisian antivirus mereka. Namun untuk yang kurang peka contohnya di pusat-pusat Internet ataupun di rumah, ancaman virus terhadap pengguna masih ada. Hal ini boleh menyebabkan kerosakan yang teruk pada perisian mahupun perkakasan komputer.

Disebabkan hal yang demikianlah EMVS ini dibangunkan untuk membolehkan pengguna terutamanya pengguna bergerak membawa sistem ini ke mana-mana sahaja dan seterusnya dapat mengoptimumkan keselamatan komputer serta fail-fail pengguna itu sendiri.

1.3 Objektif Sistem

Sistem yang dibangunkan ini adalah satu sistem pengimbas virus emel mudah alih yang akan digunakan oleh pengguna terutamanya di Fakulti Sains Komputer dan Teknologi Maklumat Universiti Malaya Kuala Lumpur. Di dalam memenuhi dan mencapai matlamat-matlamat pengguna sistem ini, beberapa objektif utama telah digariskan. Objektif tersebut adalah:

- I. Memastikan setiap komputer yang digunakan bebas daripada ancaman virus.
- II. Menghapuskan sebarang virus yang terdapat pada komputer pengguna.
- III. Melaporkan sebarang ancaman virus yang terdapat pada komputer pengguna.
- IV. Mengurangkan kos dan masa penyelenggaraan akibat serangan dan ancaman virus.
- V. Memperbaiki kelemahan yang terdapat pada EMVS V1.0.

1.4 Skop Projek

Skop projek ini boleh dibahagikan kepada beberapa bahagian. Antara bahagian-bahagian yang terkandung di dalam skop projek ini ialah skop sistem, sasaran pengguna dan bahasa yang akan digunakan.

1.4.1 Skop Sistem

Terdapat tiga modul yang akan dibangunkan dalam sistem ini. Setiap modul ini akan menjalankan fungsinya yang tertentu. Antara modul-modul tersebut adalah :

- i. Scan
- ii. Reports
- iii. Options
- iv. Update

1.4.2 Sasaran Pengguna

Skop Email Virus Scanner (EMVS) ini adalah menjadi sasaran kepada pengguna bergerak yang biasanya menggunakan banyak komputer samaada di universiti, pusat-pusat Internet mahupun di rumah. Sasaran pengguna untuk sistem ini ialah :

1. Pelajar

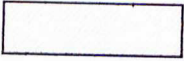
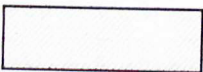





Sistem ini adalah sesuai digunakan oleh pelajar yang sering betukar-tukar tempat dalam menggunakan komputer tidak kiralah samaada di universiti, pusat Internet mahupun di rumah.

1.4.3 Bahasa Yang Digunakan

Bahasa tabii yang akan digunakan untuk Email Virus Scanner (EMVS) ini ialah Bahasa Inggeris. Bahasa ini dipilih adalah untuk memudahkan penggunaan system ini. Ini bermaksud pengguna tidak akan mudah terkeliru dengan maksud perkataan atau ayat yang digunakan berbanding jika menggunakan bahasa lain. Hal ini demikian juga adalah untuk bersaing dengan perisian-perisian lain di pasaran yang kebanyakannya menggunakan Bahasa Inggeris.

1.5 Had Sistem

Email virus scanner (EMVS) mi mempunyai beberapa kekangan yang tidak dapat dielakkan. Antara kekangan tersebut ialah :

	Aktiviti	Bulan						
		Jul	Ogos	Sept	Okt	Nov	Dec	Jan
	Perancangan sistem							
2	Analisa sistem							
3	Rekabentuk sistem							
4	Implementasi sistem							
5	Integrasi dan pengujian							
6	Penyelenggaraan sistem							
7	Dokumentasi system							

Jadual 1.1 : Garis Masa Pembangunan Projek

- ❖ Modul-modul sistem yang diuji ditentukan kesesuaiannya dengan perkakasan sistem.

VI. FASA PENYELENGGARAAN

- ❖ Sebarang perubahan yang perlu dilakukan akan dilaksanakan selepas fasa pengujian.

VII. FASA DOKUMENTASI

- ❖ Menyediakan manual pengguna, (panduan pengguna) supaya mereka dapat memahami cara sistem digunakan.
- ❖ Menyediakan format persembahan bagi menerangkan sistem yang telah siap dibangunkan secara ringkas dan padat kepada penyelia dan moderator.
- ❖ Menyediakan laporan projek yang lengkap

I. FASA KAJIAN LITERASI

- ❖ Mengumpul segala maklumat yang diperlukan untuk membangunkan sistem yang dirancang.
- ❖ Menyediakan penjadualan projek

II. FASA ANALISA SISTEM

- ❖ Maklumat yang diperolehi semasa fasa kajian literasi di analisa, dan disimpan untuk kegunaan projek pada fasa seterusnya.
- ❖ Objektif sistem ditentukan.
- ❖ Memilih dan menentukan model pembangunan yang akan digunakan.

III. FASA REKABENTUK SISTEM

- ❖ Merekabentuk format skrin utama
- ❖ Merekabentuk paparan untuk setiap modul
- ❖ Membina carta struktur sistem

IV. FASA PEMBANGUNAN SISTEM

- ❖ Mempelajari arahan serta fungsi-fungsi yang lebih kompleks di dalam perisian Visual Basic
- ❖ Melaksanakan pembangunan sistem yang telah ditentukan semasa fasa rekabentuk sistem.

V. FASA PENGUJIAN SISTEM

- ❖ Sistem yang telah dibangunkan diuji dari semasa ke semasa sepanjang tempoh fasa pelaksanaan

Sistem yang dibangunkan mestilah bebas daripada kesalahan. Oleh yang demikian percubaan adalah diperlukan.

❖ Mudah dan konsisten

Sistem mestilah menggunakan aturcara program yang mudah agar penyelenggaraan senang dilakukan. Jaminan mesti dibuat untuk mengelakkan sistem daripada mengganggu kestabilan pelayan.

❖ Kawalan capaian yang ketat

Sistem ini diharapkan tidak dapat dicapai oleh pengguna yang tidak berdaftar dengan terdapatnya fungsi login yang memerlukan nama pengguna dan katalaluan. Pengguna perlu memohon kepada pentadbir sistem untuk mendapatkan sistem ini. Pentadbir sistem kemudiannya akan memberikan sistem ini beserta nama pengguna dan katalaluannya sekali.

❖ Pemantauan yang berkesan

Diharapkan dengan terbinanya sistem ini nanti akan dapat mengurangkan ancaman atau serangan virus pada komputer di fakulti ini.

1.9 Penjadualan projek

Dalam membangun dan melengkapkan projek Email Virus Scanner ini, ia memerlukan perancangan yang teliti agar memenuhi objektif yang telah digariskan. Oleh itu, setiap perjalanan aktiviti dirancang dengan teliti dan dibahagikan mengikut fasa-fasa tertentu. Setiap fasa yang dirancang akan dipastikan siap mengikut keperluan pengguna dalam jangkamasa yang telah ditetapkan. Terdapat tujuh fasa keseluruhannya yang perlu dilalui semasa membangunkan sistem ini.

I Komputer Peribadi

Pemproses : Pentium II 350 MHz

Memori :32 MB

Cakera Liat: lebih daripada. 1.9 GB

b) Perisian

I. Windows 2000 sebagai sistem operasi

II. Visual Basic 6.0

1.8 Hasil Yang Dijangkakan

Sesuatu projek atau. sistem yang dijalankan atau dibangunkan mestilah mempunyai hasil jangkaannya. Hasil yang dijangkakan daripada pembangunan projek Email Virus Scanner (EMVS) ini adalah seperti berikut:

❖ Mengurangkan kerosakan

Dengan terbinanya sistem ini, diharapkan dapat mengurangkan kerosakan pada perkakasan mahupun perisian komputer.

❖ Mesra pengguna

Pengguna EMVS ini adalah di kalangan pelajar. Maka, antaramuka sistem ini mestilah membolehkan pengguna memahami cara mengendalikan dan berinteraksi dengannya.

❖ Tepat

Pengimbasan mestilah tepat untuk membolehkan pengesanan sebarang virus pada emel.

❖ Dipercayai

1. EMVS dibangunkan untuk memenuhi keperluan pengguna komputer terutamanya di Fakulti Sains Komputer dan Teknologi Maklumat Universiti Malaya.
11. Fungsi EMVS hanya tertumpu kepada pengimbasan, serta penghapusan virus mengikut pilihan pengguna.
111. Fungsi EMVS hanya dapat mengimbas dan mengesan beberapa bentuk virus sahaja.

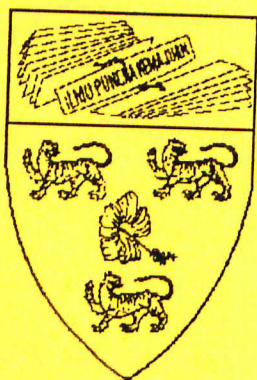
1.6 Kepentingan Sistem

Email Virus Scanner (EMVS) ini penting untuk menggalakkan penggunaan pengimbas virus pada sebarang komputer yang digunakan. Sistem ini berupaya mengimbas dan menghalang virus daripada mengancam komputer pengguna melalui pilihan-pilihan tertentu seterusnya akan melaporkan kepada pengguna tentang ancaman virus yang terdapat pada komputer mereka. Pengguna juga akan mendapat faedah daripada EMVS ini kerana mereka tidak akan lagi dibebani oleh virus-virus yang boleh mengancam serta merosakan sistem atau fail-fail pada komputer mereka. Secara keseluruhannya, EMVS ini penting untuk mengelakkan ancaman dan jangkitan daripada virus yang terdapat pada komputer mereka dan seterusnya dapat mengurangkan kos serta masa penyelenggaraan sesuatu komputer.

1.7 Keperluan Sistem

Di dalam mengimplementasikan projek Email Virus Scanner (EMVS) ini, beberapa perkakasan dan perisian telah dipilih. Antaranya ialah:

- a) **Perkakasan**



BAB 2

KAJIAN LITERASI

University of Malaya

2. KAJIAN LITERASI

2.1 Pengenalan

Kajian literasi merupakan kajian awal bagaimana sesuatu sistem dapat dibangunkan. Melalui kajian yang dijalankan ini objektif dan skop sebenarnya dapat di tentukan. Selain daripada itu melalui kajian ini juga teknologi-teknologi dapat di kenalpasti untuk di gunakan dalam pembangunan sistem.

Kajian literasi adalah penting dalam usaha untuk mendapatkan pengetahuan yang berkaitan dan dikira sebagai faktor-faktor penting untuk memenuhi objektif dan skop projek ini.

Untuk menjalankan proses kajian literasi yang membawa kepada keputusan yang boleh diguna pakai dan dipercayai terdapat beberapa langkah yang disarankan:

1. Menetapkan objektif projek yang kukuh dan senang diukur.
2. Berdasarkan objektif beberapa kajian boleh dilakukan terhadap:
 - I. Kertas kerja yang berkaitan dengan objektif yang telah ditetapkan .
 - II. Kertas kerja tentang sistem-sistem yang telah di bina yang relevan dengan objektif yang ditetapkan

- III. Rujukan ke atas internet, majalah dan lain-lain media tentang sistem yang sedia ada yang mempunyai kaitan yang rapat dengan sistem yang hendak dibangunkan.

Berdasarkan kajian literasi ini, hasil yang dijangkakan perlulah berguna dalam fasa-fasa pembangunan sistem iaitu fasa analisis, rekabentuk, implementasi, pengujian dan penyelenggaraan sistem.

Untuk tujuan projek ini, kajian literasi lebih memfokus kepada pengumpulan maklumat yang berkaitan dengan pembangunan sistem pengaturcaraan. Kajian ini perlulah lebih menjurus kepada teori pembangunan sistem dalam konteks pembinaan sistem pengimbas virus pada emel. Dengan itu proses ini akan dijangka dapat membantu dalam proses pembangunan sistem terutamanya dari segi rekabentuk dan perancangan sesebuah sistem.

2.2 Pengenalan Kepada Email Virus Scanner

2.2.1 Pengenalan Emel

Emel atau mel elektronik adalah istilah yang diberikan kepada mesej elektronik, biasanya dalam bentuk mesej teks yang mudah, yang ditaip pada sistem komputer dan dihantar melalui rangkaian komputer kepada pengguna lain yang boleh membaca emel tersebut.

Emel pada mulanya terdiri daripada pemilikan sistem emel tersebut. Sistem emel ini biasanya hanya boleh menghantar dan menerima emel dalam pejabat di mana setiap orang dibekalkan dengan perisian yang sama. Tetapi, dengan perkembangan dunia Internet kini, sebahagian pengeluar sistem emel tersebut memperkenalkan kebolehpayaan penyambungan ke Internet untuk menghantar mesej keluar daripada rangkaian setempat. Ini memerlukan antaramuka perisian yang boleh menukar mesej setempat ke bentuk yang bersesuaian untuk di hantar melalui Internet.

Sejak perkembangan Internet, pemilikan sistem menjadi kurang popular, dengan kebanyakan perniagaan beralih ke Internet untuk servis mel rangkaian setempat. Ini banyak memberi kelebihan kerana ianya lebih murah, mudah dan tidak terikat kepada pembekal tertentu serta membenarkan pengurus IT mempunyai pilihan yang lebih luas untuk aplikasi-aplikasi pelayan emel atau pelantar perkakasan yang berlainan.

2.2.2 Kebaikan Emel

Emel telah menjadi satu daripada tulang belakang penyambungan perniagaan ke Internet kerana ianya cepat dan penghantaran mesej yang ekonomikal ke mana-mana sahaja di serata dunia. Sistem emel ini juga akan mengurangkan bil telefon secara keseluruhannya kerana kos penghantaran emel melalui Internet jauh lebih murah daripada membuat panggilan telefon.

Kelebihan yang ada pada sistem emel ini telah menggalakkan lebih banyak perniagaan untuk melabur dalam pelaksanaan servis-servis emel. Emel pada keseluruhannya lebih baik berbanding cara lama yang berasaskan memo dan sistem pos.

Melalui emel, mesej boleh di hantar pada bila-bila masa ke serata dunia dengan mudah samaada kepada sekumpulan penerima atau kepada penerima tunggal tanpa perlu meninggalkan meja. Mesej boleh dilogkan, dipastikan bahawa bentuk rekod dipegang dan boleh di simpan oleh penerima. Penerima boleh mengambil mel mereka pada bila-bila sahaja meraka mahukan, daripada mana pun mereka berada.

Pengguna bergerak boleh mengambil mel mereka tanpa perlu melawat pelanggan atau ke lokasi-lokasi lain. Penerima juga boleh menerima terus mesej yang di hantar tanpa sebarang peraturan dan yang lebih penting lagi adalah kerana emel adalah mesra persekitaran di mana mesej tidak lagi memerlukan kertas atau sumber-sumber lain kecuali hanya memerlukan ruang simpanan pada cakera keras computer.

2.2.3 Pelanggan Emel

Pelanggan emel adalah aplikasi yang digunakan untuk membaca, menulis dan menghantar emel. Lebih mudah lagi jika ia dikatakan sebagai antaramuka pengguna system emel tersebut.

Pelanggan biasanya terdiri daripada kombinasi penyunting teks yang mudah, buku alamat, kabinet pemfailan dan modul komunikasi.

Penyunting teks membenarkan untuk rekaan mesej dan biasanya termasuklah pemeriksa ejaan mudah dan kemudahan pemformatan.

Keupayaan untuk membenarkan fail-fail atau dokumen-dokumen untuk dilampirkan bersama mesej juga dibolehkan. Sebagai contohnya, diagram boleh dilampirkan kepada mesej emel yang membolehkan penerima melihat perjalanan sesuatu projek dan boleh memberikan sebarang komen dengan membalas semula emel tersebut.

Buku alamat membenarkan pengguna menyimpan alamat emel dengan format yang mudah untuk mengurangkan kebarangkalian penghantaran emel dengan menggunakan alamat yang salah.

Kabinet pemfailan pula membenarkan penyimpanan mesej emel samaada yang dihantar atau diterima dan biasanya memberikan fungsi carian untuk membenarkan capaian semula mesej yang dikehendaki.

2.2.4 Pelayan Emel

Pelayan emel adalah aplikasi yang menerima emel dari pelanggan emel atau pelayan-pelayan emel yang lain. Pelayan emel biasanya terdiri daripada kawasan

penyimpanan, set peraturan yang menerangkan pengguna, senarai pengguna-pengguna dan modul-modul komunikasi.

Kawasan penyimpanan adalah tempat penyimpanan mel untuk pengguna setempat dan juga tempat penyimpanan sementara bagi mesej yang hendak di hantar ke destinasi yang lain. Ia biasanya adalah dalam bentuk pangkalan data maklumat yang mudah.

Peraturan yang menerangkan pengguna menentukan bagaimana pelayan mel bertindak balas bila menentukan destinasi mesej yang spesifik atau berkemungkinan bertindak balas terhadap penghantar mesej itu sendiri.

Senarai pengguna pula adalah pangkalan data akaun pengguna yang akan dikenalpasti dan dibahagi-bahagi di pangkalan data ini oleh pelayan mel.

Modul-modul komunikasi adalah komponen-komponen yang mengendalikan penghantaran mesej kepada dan daripada pelayan mel dan pelanggan emel yang lain. Modul-modul yang berlainan dipasang bergantung kepada kegunaan dan keperluan pelayan mel.

Seseorang yang dipanggil 'Postmaster' pula menjaga pelayan mel dan senarai akaun pengguna yang disokong olehnya.

Kebanyakan pelayan mel direka untuk beroperasi tanpa sebarang campurtangan semasa operasi biasa. Mereka menunggu untuk mesej di hantar kepada mereka

dan memprosesnya atau mengumpul mesej daripada pelayan-pelayan mel yang lain pada masa ketaktentuan.

2.2.5 Pengenalan Virus Komputer

Virus komputer adalah suatu program, sejenis kod pelaksana yang mempunyai kemampuan yang unik untuk menyalin. Seperti virus-virus biologikal, virus-virus komputer juga boleh tersebar dengan cepat dan sentiasa sukar untuk dihapuskan. Ia boleh lampirkan dirinya pada apa-apa jenis fail dan tersebar sebagai fail salinan yang dihantar dari individu kepada individu.

Selain penyalinan, sesetengah virus komputer mempunyai sesuatu lagi yang biasa: rutin kerosakan yang boleh menghantar virus 'payload'. 'payload' bukan sahaja boleh memaparkan mesej atau imej tetapi juga boleh memusnahkan fail-fail, membentuk semula cakera keras atau menyebabkan kerosakan-kerosakan yang lain. Jika virus tidak mempunyai rutin kerosakan, ia masih boleh menyebabkan masalah dengan mengambil ruang simpanan atau memori dan menurun tahap keseluruhan pelaksanaan komputer.

Beberapa tahun dahulu, kebanyakan virus merebak melalui media seperti cakera liut, tetapi kini Internet telah menyebabkan virus mempunyai mekanisme baru untuk disebarkan. Memandangkan emel kini digunakan sebagai peralatan komunikasi yang penting, virus telah merebak dengan begitu cepat melalui medium ini. Virus yang dilampirkan pada mesej emel, boleh menyerang

keseluruhan komputer organisasi dalam masa beberapa minit sahaja, menyebabkan kehilangan produktiviti serta kos baik pulih yang mahal.

Secara amnya terdapat tiga kelas utama virus :

1. **Penyerang fail.** Virus-virus ini dilampirkan pada fail-fail program, biasanya fail-fail .COM atau .EXE. Sebahagiannya boleh menyerang sebarang program di mana perlaksanaan dikehendaki termasuklah fail-fail .SYS, .OVL, .PRG dan .MNU. Apabila fail dibebani ('loaded'), virus juga akan dibebani bersama.
2. **Penyerang sistem atau boot-record.** Virus-virus ini menyerang kod pelaksana yang dijumpai di kawasan sistem di dalam cakera. Ia dilampirkan kepada sektor boot DOS atau Master Boot Record pada cakera keras.
3. **Virus-virus makro.** Ini adalah antara virus-virus biasa yang menyebabkan kerosakan yang kecil. Virus makro akan menyerang aplikasi Microsoft Word dan biasanya akan memasukkan perkataan-perkataan atau frasa-frasa yang tidak dikehendaki.

2.2.6 Kaedah Dan Teknik Pengesanan Virus Komputer

Teknik pengesanan virus adalah satu program yang akan mencari virus pada fail atau pada rekod permulaan sistem komputer. Untuk membolehkan proses pengesanan virus berlaku seperti yang dikehendaki dan dapat mengesan kehadiran virus yang baru, pengaturcara telah melakukan spesifikasi program yang mampu mengesan virus yang terbaru. Pengesanan virus hanya boleh dilakukan di tempat ia disyaki wujud.

Berikut adalah teknik-teknik pengesanan virus yang sering digunakan oleh produk antivirus yang berada dalam pasaran:

I. Teknik penyemakan secara bait perkataan

Teknik pertama yang digunakan untuk proses pengesanan virus oleh kebanyakan perisian antivirus ialah dengan menggunakan konsep pertahanan secara carian dan padanan bait perkataan. Melalui teknik pengesanan ini, proses pencarian akan dilakukan pada setiap bait fail program untuk mencari jujukan bait yang menyamai pengenalan yang sama seperti virus.

Jika proses telah dapat mengesan kewujudan jujukan bait yang disyaki, maka laporan akan dikeluarkan bagi memberitahu fail tersebut telah mengandungi virus. Namun dengan menggunakan teknik ini proses pengesanan menjadi lambat kerana proses terpaksa memperuntukkan jumlah masa yang besar untuk melakukan pemeriksaan serta penentuan jujukan bait yang disyaki untuk setiap bait dalam fail.

Pengkaji antivirus seterusnya telah mempertingkatkan kecekapan pelaksanaan teknik pengesanan virus supaya tempoh masa proses penyemakan fail akan menjadi lebih menjimatkan. Pengkaji antivirus melakukan perubahan pada penyemakan bait perkataan dengan menambah keupayaan penyemakan menggunakan carian secara kod unik perkataan yang lebih terperinci. Teknik penambahan kod unik perkataan ini pada pengesanan virus dinamakan teknik imbasan dan penyemakan. Teknik ini menggunakan kod unik perkataan yang mudah bagi pengenalan virus semasa menjanakan aktiviti pengesanan.

II. Teknik penyemakan berdasarkan pengenalan unik

Teknik penyemakan seterusnya yang sering digunakan untuk proses pengesanan ialah penyemakan berasaskan pengenalan virus. Apabila aktiviti semakan berada dalam keadaan aktif, ia akan memeriksa setiap fail yang diberikan padanya dan di dalam kes EMVS ini, proses penyemakan akan memeriksa fail yang terkandung di dalam direktori tertentu.

Pengenalan adalah bekas identiti virus komputer yang mana ia merupakan pengenalan unik yang terdapat pada setiap virus yang wujud pada masa kini. Penyemakan virus berdasarkan pengenalan ini akan melakukan pemeriksaan pada semua bahagian jadual format fail untuk mencari kod perkataan yang berpadanan dengan pengenalan diri virus yang tersimpan di dalam pangkalan data pengenalan unik virus.

Apabila semakan telah dapat mengenalpasti pengenalan virus yang berpadanan, ia akan menentukan jenis virus berdasarkan nama virus tersebut. Proses penyemakan juga mampu untuk mengesan virus sebelum virus berpeluang untuk merebak dan dengan kemampuan ini ia dapat mengurangkan peluang berlakunya penyebaran virus dalam sistem perkomputeran.

2.2.7 Pemilihan Teknik Pengesanan Virus Komputer

Dalam system EMVS teknik imbasan melalui penentuan pengenalan unik virus digunakan. Faktor –faktor pemilihan teknik ini adalah seperti dibawah.

2.2.7.1 Kelebihan Teknik Penentuan Pengenalan Unik Virus

Teknik ini telah dipilih untuk digunakan dalam EMVS disebabkan faktor-faktor yang berikut:

I. Mudah digunakan dan difahami

Pengguna komputer akan dapat menggunakan aplikasi teknik ini melalui antaramuka yang tersedia. Proses pengesanan dan penghapusan fail virus dapat dilakukan secara manual oleh pengguna melalui pelaksanaan aplikasi pengesanan yang mudah. Selain daripada program aplikasi yang menggunakan teknik penentuan pengenalan virus juga mudah dibangunkan oleh pengaturcara.

II. Ketepatan

Enjin pengesanan adalah pengesan yang boleh dipercayai untuk mengesan kod virus yang telah menjangkiti sistem komputer. Kebanyakan enjin pengesanan dibangunkan bersama-sama pangkalan data yang mengandungi senarai keseluruhan kod virus yang terdapat di dunia ini. Pangkalan data kod virus ini akan sering dikemaskini dan diselenggara oleh syarikat yang tertentu.

III. Kecekapan

Kelengkapan pengesanan mampu melakukan proses carian dengan cekap dan melakukan penjanaan aplikasi pengesanan kod virus dengan aktif. Proses

pengesanan akan menjejaki kod virus satu demi satu padanan tanpa meninggalkan satu bit yang terdapat pada kod segmen.

2.2.7.2 Kelemahan Teknik Penentuan Pengenalan Unik Virus

Terdapat dua kelemahan menggunakan teknik penentuan pengenalan unik virus ini iaitu:

I. Berlaku keadaan positif palsu

Kejadian positif palsu berlaku apabila proses pelaksanaan mendapati tiada jangkitan virus ditemui dalam program utama. Tetapi laporan itu sebenarnya tidak menunjukkan keadaan yang sebenar kerana semasa proses pelaksanaan berlaku, bait unik bagi virus telah disertakan bersama-sama program utama. Maka, kewujudan virus tidak dapat dikenalpasti dengan tepat. Situasi positif palsu ini lebih dikenali sebagai kesalahan jenis 1.

II. Berlaku keadaan negatif palsu

Keadaan negatif palsu berlaku apabila proses pelaksanaan melaporkan bahawa jangkitan virus telah ditemui pada program utama walaupun tiada persamaan padanan bait virus ditemui pada fail tersebut. Laporan daripada proses pelaksanaan tersebut sebenarnya silap kerana proses pelaksanaan tidak dapat menemui kewujudan virus pada program.

2.2.8 Analisa Masalah

Teknik pengesanan yang digunakan dalam system EMVS ini ialah teknik penentuan penenalan unik virus. Fail yang telah dijangkiti oleh virus akan dihapuskan oleh sistem dan fail tersebut tidak akan disimpan di dalam direktori sistem komputer.

Sistem EMVS ini membawa konsep yang sama seperti perisian antivirus yang terdapat di pasaran kini seperti Norton Antivirus, PC-Cilin dan McAfee. Perisian antivirus ini menyediakan fungsi-fungsi seperti penentuan fail yang akan diimbas, mesej laporan samada fail bebas daripada virus atau telah dijangkiti dan proses penghapusan fail yang telah dijangkiti virus.

2.3 Kaedah Kajian

Secara umumnya, pembangunan sistem tidak akan sempurna jika tiada pengumpulan dan penyelidikan maklumat tentang sistem yang akan dibangunkan. Oleh hal demikian, maklumat adalah penting untuk membantu sistem mencapai matlamat dan objektif yang telah disasarkan dengan tepat. Maklumat boleh diperolehi daripada pelbagai sumber dan setiap sumber memberikan maklumat yang berbeza-beza. Ini penting bagi mengetahui sejauh mana keperluan terhadap perisian yang akan dibangunkan dan memastikan kajian tentang sistem yang akan dibangunkan adalah menyeluruh. Pelbagai

kaedah telah digunakan di dalam menjalankan kajian ilmiah ini. Di antara kaedah yang digunakan ialah :

- I. Kaedah pengumpulan data
- II. Kaedah penulisan

2.3.1 Kaedah Pengumpulan Data

Di dalam usaha mengumpulkan data, beberapa pendekatan telah digunakan. Antaranya ialah:

I. Perbincangan dengan penyelia

Sebelum melaksanakan kajian keperluan sistem, perbincangan dengan penyelia telah dilakukan. Ini bertujuan untuk mendapatkan nasihat dan pandangan serta gambaran tentang sistem yang akan dibangunkan. Perbincangan dengan penyelia adalah penting untuk mendapatkan dan mengetahui tahap sebenar sistem yang akan dibangunkan kelak. Perbincangan ini bertindak sebagai satu garis panduan dalam membangunkan sesuatu sistem.

II. Dokumentasi

Kaedah ini adalah satu kaedah pengumpulan data, melalui kajian dan analisa terhadap dokumen-dokumen yang ada kaitan dengan kajian yang dilakukan.

Dokumen-dokumen yang diperolehi adalah samada melalui koleksi peribadi rakan-rakan atau melalui bahan rujukan daripada perpustakaan.

III. Melayari Internet

Maklumat juga diperolehi daripada Internet yang merupakan sumber informasi yang paling penting dan popular. Antara maklumat yang diperolehi ialah maklumat tentang sistem yang sedang dikaji dan maklumat-maklumat yang berkaitan dengannya.

Penggunaan enjin pencari ini adalah berdasarkan kata kunci yang ditaip masuk yang merupakan maklumat yang ingin dicari. Tujuan penggunaannya ialah:

- Untuk mendapatkan maklumat berkenaan sistem pengimbas virus pada emel yang telah wujud dirangkaian iaitu dikaji berdasarkan ciri-ciri seperti antaramuka dan sejauh mana sistem itu dapat membantu pengguna. Kata kunci yang digunakan ialah *Email Virus Scanner*.
- Untuk memperolehi maklumat tentang langkah-langkah yang digunakan dalam pembangunan sistem seperti kriteria dan langkah yang diperlukan. Enjin pencari digunakan untuk mendapatkan panduan yang penting yang terdapat dalam Internet dan kata kunci yang digunakan ialah *System Analyst and Design*.

IV. Pemerhatian

Melalui kaedah ini meklumat-maklumat yang diperlukan dikumpul melalui pengamatan dan penelitian secara langsung terhadap subjek kajian yang dimaksudkan. Di antara skop yang dilihat termasuklah realiti operasi sistem dan kelemahan-kelemahan sistem.

V. Temubual atau temuramah

Kaedah ini dilakukan dengan menemubual pentadbir pelayan mel dan beberapa pembantu makmal di Fakulti Sains Komputer dan Teknologi Maklumat Universiti Malaya, Kuala Lumpur berpandukan soalan-soalan yang disediakan mahupun soalan secara spontan. Ini bertujuan untuk mendapatkan gambaran sebenar tentang operasi sistem, keperluan sistem dan apa-apa yang berkaitan dengannya. Maklumat yang diperolehi dapat dijadikan panduan dan digunakan sebagai kriteria yang perlu ada dalam sistem yang akan dibangunkan. Selain daripada itu, hasil daripada temubual ini juga dapat membantu dalam menyediakan modul-modul yang berkaitan dan relevan untuk dimasukkan dalam sistem ini.

2.3.2 Kaedah Penulisan

Kaedah-kaedah penulisan berikut telah digunakan untuk menyediakan dokumentasi:

I. Kaedah analisa

Menganalisa semula dan menghuraikan sendiri segala maklumat dan data yang diperolehi ke dalam format yang lebih ringkas serta lebih menepati kehendak dan tujuan subjek ini.

II. Kaedah perbandingan

Kaedah ini dilakukan dengan membuat kesimpulan dan keputusan kajian melalui perbandingan antara dua atau lebih maklumat yang diperolehi. Kaedah perbandingan yang difokuskan di sini ialah membandingkan sistem-sistem yang sedia ada dengan sistem yang akan dibangunkan.

2.4 Analisis Sistem Sedia Ada

Pembangunan suatu sistem adalah bergantung kepada identiti sistem tersebut iaitu ‘apakah tujuan ianya dibina dan dilancarkan?’ dan ‘siapakah golongan sasaran serta skop maklumat yang hendak disampaikan?’. Bagi pembangunan Email Virus Scanner (EMVS) ini, ia memerlukan rekabentuk yang menarik, mudah difahami dan digunakan, tahap keberkesanan perlaksanaan yang baik dan dapat mencapai objektif yang telah ditetapkan.

Bagi memastikan sistem yang akan dihasilkan nanti lebih berkualiti, maka EMVS V1.0 perlu dikaji.

EMVS V1.0

System ini dibangunkan oleh pelajar FSKTM, UM beberapa semester yang lalu. System ini bersaiz kecil dan boleh disimpan dalam satu disket sahaja. Ia hanya boleh mengesan dua jenis virus sahaja iaitu *Love & Trojan*.

Terdapat tiga modul yang telah dibangunkan dalam system ini iaitu :

❖ **Scan**

Ia menyediakan pilihan direktori kepada pengguna untuk memilih direktori mana yang ingin diimbas. Tetapi system ini tidak dapat mengimbas direktori mengikut pilihan pengguna. Sebaliknya ia akan mengimbas keseluruhan pemacu tersebut.

❖ **Reports**

Modul ini membolehkan rekod pengimbasan virus seperti Deleted Item, Activity log dan Virus List dilihat kembali. Namun system ini tidak membenarkan pengguna untuk menyimpan rekod tersebut untuk melihatnya kembali pada masa akan datang.. Ia akan hilang sebaik sahaja pengguna menamatkan system.

❖ **Options**

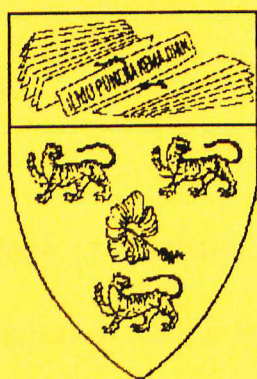
Modul ini membolehkan pengguna menentukan samada system ini akan secara automatic diaktifkan atau pun tidak setiap kali system pengoperasian bermula.

2.5 Perbandingan Sistem

Dalam membangun system EMVS V1.1 ini, beberapa kelemahan dalam EMVS V1.0 akan diperbaiki. System yang dibangun ini diharap akan dapat mengatasi kelemahan tersebut.

Antara perubahan dan penambahan yang telah dirancang untuk dilaksanakan adalah :

- ❖ Penambahan *option* di antaramuka log in untuk membenarkan pengguna menukar user id atau password. Ini akan memudahkan pengguna untuk mengingat user id dan password.
- ❖ Oleh kerana EMVS V1.0 hanya boleh mengesan dua jenis virus sahaja, maka dalam EMVS V1.1, beberapa jenis virus emel yang popular dan merbahaya dirancang akan ditambah. Ini akan meningkatkan keupayaan system untuk mengesan virus.
- ❖ Membolehkan pengguna memilih direktori yang ingin diimbas dan sistem ini akan mengimbas direktori yang dipilih oleh pengguna sahaja.
- ❖ Membolehkan rekod pengimbas virus serta penghapusan virus yang terdahulu disimpan dan seterusnya mampu untuk dilihat kembali.
- ❖ Mengubah beberapa fungsi supaya lebih mesra pengguna.
- ❖ Penambahan modul *Up Date* untuk membolehkan pengguna mengemaskini virus-virus supaya lebih terkini.
- ❖ Jika dalam EMVS V1.1 ia disimpan dalam disket, tetapi EMVS V1.1 ini akan disimpan di dalam CD.



BAB 3 METODOLOGI

BAB 3 METODOLOGI

3.1 Metodologi Sistem

Pembangunan sesebuah projek perlu melalui beberapa proses tertentu sebelum ia dapat menghasilkan keputusan atau output akhir yang dikehendaki. Turutan perjalanan bagi proses-proses ini perlu ditunjukkan dengan menggunakan salah satu daripada metodologi pembangunan sistem yang ada di dalam bidang kejuruteraan perisian.

Metodologi adalah merupakan satu set panduan lengkap yang mengandungi model-model, kemudahan peralatan (tool) dan teknik-teknik yang perlu diikuti dalam melaksanakan setiap aktiviti yang terdapat dalam kitar hayat pembangunan suatu sistem. Metodologi yang bersesuaian dan tepat memainkan peranan yang penting bagi penghasilan dan persembahan produk perisian atau sistem yang betul dan boleh dipercayai.

Terdapat dua objektif metodologi pembangunan sistem iaitu:

1. Memastikan semua individu yang terlibat dalam projek sedia maklum dengan tujuan, progres dan masalah bagi suatu pembangunan perisian
2. Membahagikan pembangunan projek kepada beberapa langkah dan fasa-fasa yang boleh diuruskan dengan penyempurnaan berisyarat oleh hasil akhir yang spesifik.

3.1.1 Faedah Menggunakan Metodologi

Di antara faedah-faedah menggunakan metodologi ialah:

I. Meningkatkan produktiviti

Produktiviti projek akan meningkat dan memberikan hasil yang menepati kehendak pengguna serta pemilik sistem perisian.

II. Memperbaiki kualiti

Kualiti sistem perisian yang dihasilkan akan menjadi lebih baik dan bermutu. Produk dihasilkan dengan teratur dan segala proses berjalan dengan lancar.

III. Dokumentasi yang lebih baik

Dokumentasi sistem yang dipersembahkan merupakan yang lebih baik berbanding dokumentasi tanpa menggunakan metodologi. Semua proses dapat didokumenkan dan diterangkan secara jelas dan terang.

IV. Penyelenggaraan jangka hayat dikurangkan

Penggunaan metodologi juga dapat mengurangkan penyelenggaraan jangka hayat suatu sistem perisian. Masalah dan kerumitan dapat dikurangkan sejak dari peringkat awal lagi dan ini telah memberi kelebihan kepada semua pihak yang terlibat untuk menyelenggarakan dan mengendalikan sistem tersebut.

V. Pengurangan kos

Selain itu, beban kos dapat diringankan kerana masa yang diambil untuk menyiapkan suatu sistem lebih singkat dan masalah yang dikesan dari peringkat awal memerlukan kos yang lebih rendah untuk diperbaiki berbanding kos yang diperlukan jika ada masalah dikesan diperingkat perlaksanaan sistem.

3.2 Jenis Metodologi

Terdapat pelbagai kaedah metodologi yang boleh digunakan dalam pembangunan sistem. Di antaranya adalah seperti Metodologi Air Terjun, Model V, Model Transformasi, Model Pemprototaipan, alatan CASE dan RAD serta banyak lagi.

Walaubagaimanapun, apa yang penting di dalam peringkat ini ialah mengenalpasti tugas utama yang perlu dilakukan oleh perəkabentuk dan bagaimana untuk mengendalikan sesuatu pembangunan sistem dengan baik. Dalam pembangunan sistem Email Virus Scanner (EMVS), pendekatan yang di ambil ialah menggunakan Model Air Terjun dengan Pemprototaipan.

3.3 Pilihan Metodologi

Metodologi yang digunakan dalam membangunkan Sistem Email Virus Scanner ialah model air terjun dengan pemprototaipan. Ini adalah kerana keperluan sistem untuk projek ini boleh didefinisikan melalui kajian dan penyelidikan daripada sistem, alatan dan teknologi yang ada. Di samping itu, melalui model air terjun dengan prototaip, dokumentasi setiap peringkat proses pembangunan boleh disiapkan dengan mudah kerana model ini dipecahkan kepada peringkat-peringkatnya dengan jelas dan ianya akan lebih mudah difahami dan lebih berstruktur. Setiap fasa seperti spesifikasi keperluan, rekabentuk sistem, perlaksanaan, pengujian dan seterusnya ditunjukkan

secara berasingan tetapi berturutan. Selepas setiap satu fasa telah dilakukan, pembangunan akan diteruskan pula ke fasa-fasa yang berikutnya.

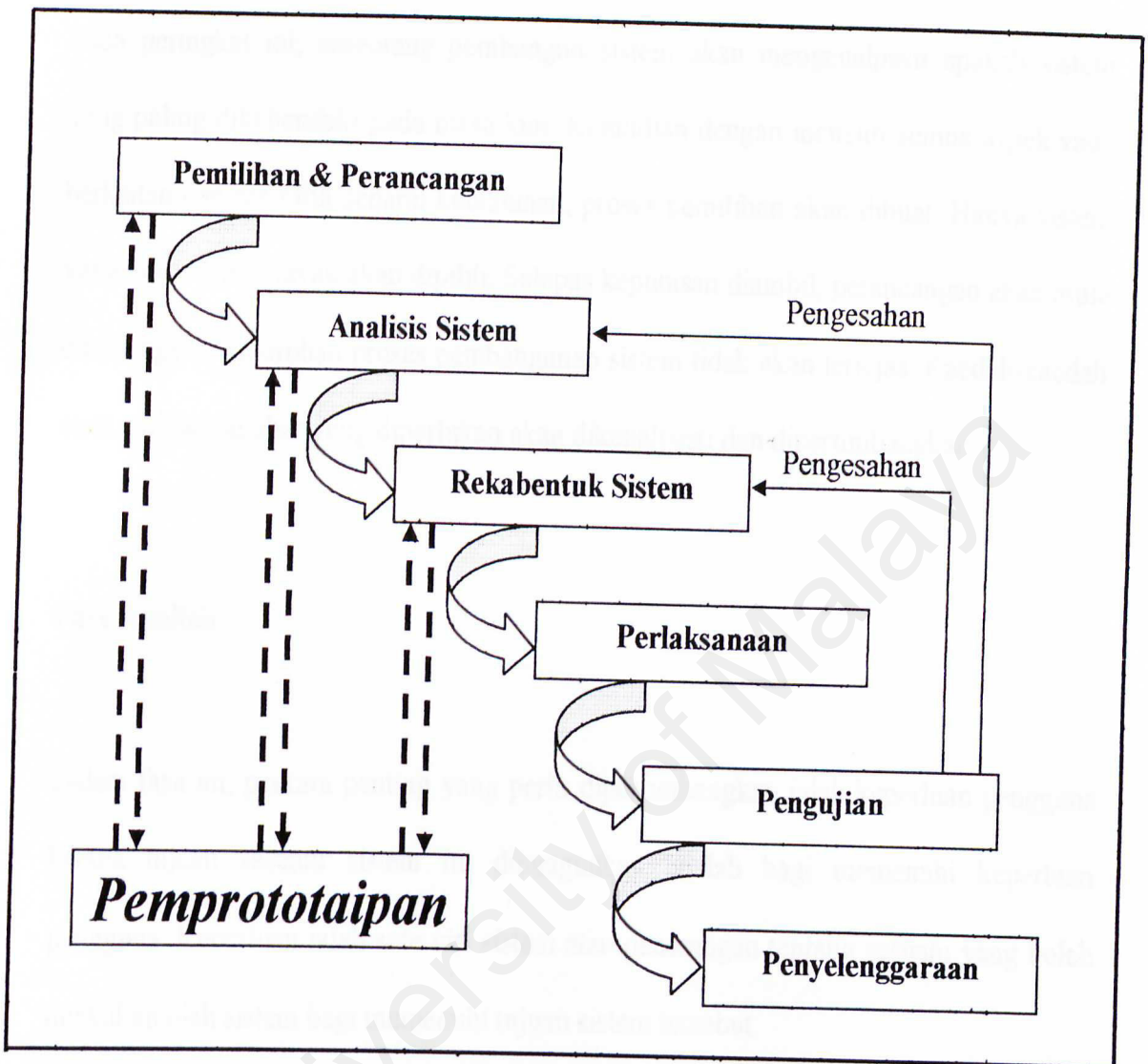
Pemprototaipan kebiasaannya merupakan proses iteratif. Prototaip ialah sebahagian kecil daripada sistem yang dibangunkan untuk menguji beberapa aspek sistem yang dicadangkan. Sebagai contoh, pembangun boleh membina model prototaip dan menilainya daripada maklumbalas pengguna. Proses itu berulang dalam fasa yang sama sehingga kedua-dua pihak berpuashati. Selepas itu, pembangun akan meneruskan kepada fasa berikutnya dan mengulangi proses pemprototaipan.

Antara kebaikan daripada gabungan kedua-dua model ini untuk projek yang sedang dibangunkan ialah:

- ❖ Model ini boleh menghasilkan sesuatu sistem yang berkualiti tinggi. Ini kerana setiap fasa pembangunan memerlukan penumpuan yang teliti bagi mengelakkan dari berlakunya sesuatu masalah yang menyebabkan sistem tersebut terpaksa dibangunkan semula.
- ❖ Model ini dapat mengesan setiap permasalahan dari peringkat awal lagi. Ini membolehkan setiap masalah dan kekurangan pada sistem tersebut diperbaiki.
- ❖ Keperluan serta perkhidmatan pengguna yang tidak dimasukkan dalam sistem juga boleh dikesan pada peringkat awal pembangunan sistem.
- ❖ Model ini merupakan paradigma yang biasa dan ia mempunyai banyak fasa pembangunan. Oleh itu, sejak dahulu lagi ia digemari dan banyak digunakan oleh para pembangunan sistem.

- ❖ Pelaksanaan kerja secara berperingkat yang diamalkan dalam model ini juga dapat mewujudkan pembahagian kerja secara sistematik dan berkesan.
- ❖ Model ini juga sangat terkenal di kalangan para pembangun sistem kerana ia sangat mudah difahami dan dilaksanakan. Ia sekaligus dapat memastikan segala keperluan dalam pembangunan sistem ini akan dipenuhi.
- ❖ Fungsi-fungsi sistem yang sukar digunakan oleh pengguna boleh dikesan dan diperbaiki. Ia dapat memastikan pembangun membangunkan sistem yang betul mengikut spesifikasi. Ini juga membolehkan pembangun melakukan kawalan verifikasi untuk kualiti pelaksanaan
- ❖ Pemprototaipan adalah perlu untuk verifikasi dan pengesahan, di mana verifikasi memastikan bahawa setiap fungsi berfungsi dengan betul dan manakala pengesahan pula memastikan bahawa sistem itu telah melaksanakan kesemua keperluan dalam spesifikasi.

3.4 Struktur Model



Rajah 3.1 Model Air Terjun dengan Pemprototaipan

3.4.1 Penerangan Setiap Fasa Dan Aktiviti Yang Terlibat

Fasa-fasa yang terlibat di sepanjang kitar hayat pembangunan sistem metodologi model air terjun dengan prototaip adalah :

Fasa Pemilihan dan Perancangan

Pada peringkat ini, seseorang pembangun sistem akan mengenalpasti apakah sistem yang paling dikehendaki pada masa kini. Kemudian dengan meneliti semua aspek yang berkaitan dan memilih senarai keutamaan, proses pemilihan akan dibuat. Hanya sistem yang benar-benar layak akan dipilih. Selepas keputusan diambil, perancangan akan mula dibuat agar keseluruhan proses pembangunan sistem tidak akan terjejas. Kaedah-kaedah serta sumber-sumber yang diperlukan akan dikenalpasti dan dipertimbangkan.

Fasa Analisis

Dalam fasa ini, perkara penting yang perlu dipertimbangkan ialah keperluan pengguna kerana tujuan sesuatu sistem itu dibangunkan adalah bagi memenuhi keperluan pengguna. Keperluan ialah satu ciri sistem atau penerangan tentang sesuatu yang boleh dilakukan oleh sistem bagi memenuhi tujuan sistem tersebut.

Selain itu dalam fasa ini juga, segala bahan dan maklumat yang berkaitan dengan projek telah dikumpulkan untuk dianalisa. Bahan-bahan tersebut adalah seperti maklumat bercetak (buku-buku, majalah, akhbar), contoh sistem yang telah sedia ada, dan lain-lain lagi. Kesemua bahan ini akan dianalisa dan dikaji dari segenap aspek. Sebagai contoh, sistem yang telah ada akan turut dianalisa untuk mengetahui apakah kelebihan dan kelemahan setiap satunya. Dengan cara ini, satu sistem yang lebih baik mungkin dapat dibangunkan.

Fasa Rekabentuk

Fasa rekabentuk ini terbahagi kepada 2 iaitu rekabentuk secara logikal dan juga rekabentuk secara fizikal. Dalam fasa rekabentuk logikal, semua rekabentuk sistem yang telah dipilih dalam fasa analisa untuk dibangunkan akan diterang secara logik terlebih dahulu tanpa bergantung kepada mana-mana sistem yang berasaskan komputer. Fasa rekabentuk fizikal pula melibatkan sistem yang akan mula dibangunkan menggunakan teknologi dan perisian yang sedia ada. Dalam fasa rekabentuk ini juga terdapat 3 proses rekabentuk utama iaitu rekabentuk struktur, rekabentuk antaramuka pentadbir dan pengguna serta rekabentuk pangkalan data. Ketiga-tiga proses perlu dilakukan agar keseluruhan rekabentuk sistem tidak terjejas.

Fasa Perlaksanaan

Fasa ini mengandungi beberapa sub aktiviti iaitu proses pengkodan, pengujian, dan pemasangan. Proses ini dilakukan secara berperingkat berpandukan bahan dan sumber yang telah dikumpulkan. Dalam proses pengkodan ini, sistem akan dibangunkan menggunakan bahasa pengaturcaraan yang telah dipilih.

Fasa pengujian

Setelah proses pengkodan selesai, maka sistem yang telah dibangunkan itu akan diuji tahap keberkesanannya untuk memastikan ia tidak mempunyai sebarang ralat dan sistem

yang telah disiapkan adalah seperti mana yang telah dirancang serta tidak memberikan sebarang kesan dan implikasi kelak. Oleh itu, sebarang ralat dan masalah yang dapat dikesan akan segera diperbaiki.

Fasa Penyelenggaraan

Proses penyelenggaraan merupakan peringkat terakhir di dalam fasa pembangunan sistem ini. Ia dilakukan apabila sistem telah lengkap dibangunkan dan boleh dicapai oleh pengguna. Penyelenggaraan ini boleh dilakukan dari masa ke semasa mengikut keperluan sesuatu sistem.

3.4.2 Pemprototaipan

Prototaip boleh dilakukan melalui 2 cara iaitu :

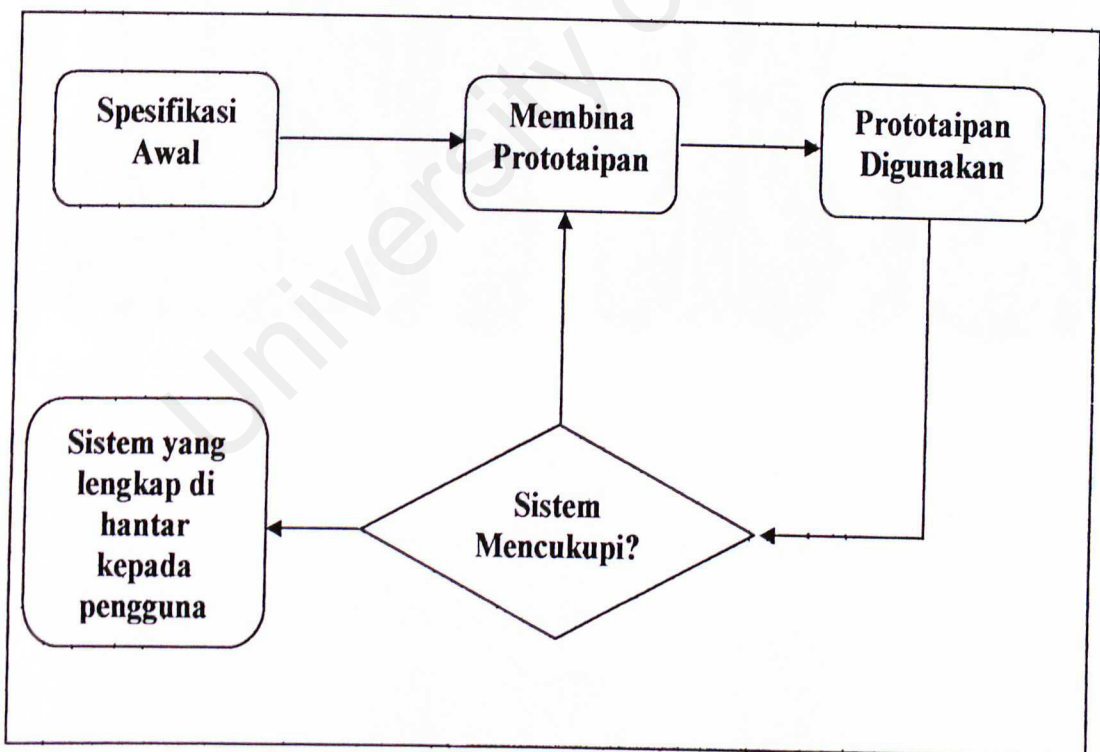
I. 'Evolutionary'

Prototaip jenis ini merujuk kepada idea membangunkan implementasi pada peringkat awal, mendedahkan proses implementasi ini kepada pengguna bagi mendapatkan komen mereka dan kemudiannya langkah ini diulangi sehinggalah suatu sistem yang lengkap telah berjaya dibangunkan.

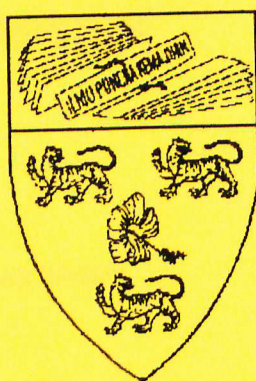
II. 'Throw-away'

Prototaip jenis ini pula menekankan spesifikasi agar output prototaip yang dilakukan merupakan spesifikasi–spesifikasi lain sistem yang belum dikenalpasti

Bagi projek ini, Prototaip 'Evolutionary' akan digunakan kerana keperluan-keperluan sistem yang diinginkan telahpun diketahui dan hanya memerlukan komen pengguna bagi memastikan sistem yang dibangunkan benar-benar memenuhi keperluan mereka.



Rajah 3.2 : Prototaip 'Evolutionary'



BAB 4 ANALISIS SISTEM

BAB 4 ANALISIS SISTEM

4.0 Spesifikasi Keperluan Sistem

Spesifikasi keperluan sistem terbahagi kepada 2 jenis iaitu keperluan fungsian dan keperluan bukan fungsian. Keperluan fungsian menyatakan tentang fungsi-fungsi yang ditawarkan oleh sistem, bagaimana sistem bertindak balas terhadap sesuatu input dan juga cara kelakuan sistem dalam keadaan itu. Keperluan bukan fungsian pula merujuk ciri-ciri lain yang perlu ada pada sistem serta had-had atau halangan ('constrain') terhadap fungsi yang ditawarkan oleh sistem. Ini termasuklah had-had yang wujud pada proses pembangunan sistem dan had masa.

4.1 Keperluan Fungsian

Keperluan fungsian menceritakan fungsi utama yang terdapat di dalam sistem yang mana ia memenuhi keperluan yang diinginkan oleh pengguna. Setiap keperluan ini menerangkan interaksi antara sistem dengan persekitarannya atau dalam maksud yang mudah ia menghuraikan sifat-sifat sesebuah sistem.

Keperluan fungsian adalah harapan pengguna terhadap sistem yang ingin dibangunkan, di mana keperluan ini menggariskan fungsi utama sistem tersebut, menghuraikan tentang

sifat-sifat sesebuah sistem serta interaksi antara sistem dengan persekitarannya. Malah, keperluan fungsian juga menghuraikan tentang bagaimana sistem harus lakukan apabila diberi arahan tertentu.

Terdapat 1 modul utama yang dirancang di dalam keperluan fungsian ini. Modul tersebut ialah Modul Pentadbir.

Modul Pentadbir

Modul pentadbir ini merupakan modul yang dikhaskan untuk pentadbir ataupun seseorang pembangun sistem untuk melakukan proses kemaskini terhadap sistem untuk aplikasi itu digunakan. Oleh itu modul ini hanya boleh dicapai oleh pentadbir ataupun pembangun sistem sahaja. Modul ini turut mempunyai sub modul. Sub modul bagi pentadbir pula ialah:

i. Login untuk sistem

Modul ini hanya membenarkan pembangun sistem atau orang yang dibenarkan sahaja untuk memasuki ruang pentadbir. Terdapatnya ciri-ciri keselamatan yang menghalang pengguna yang tidak sah daripada memasukinya.

ii. Mengubah dan menambah ciri-ciri sistem yang baru

Modul ini akan membolehkan seseorang pembangun sistem membuat sebarang pengubahsuaian dan penambahan aplikasi pada sistem mengikut keperluan dan dalam satu masa tertentu.

iii. Memadam ciri-ciri pada sistem

Modul ini membolehkan pembangun sistem memadam ciri-ciri sistem yang dirasakan tidak diinginkan dan tidak diperlukan lagi.

iv. Mengemaskini sistem

Modul ini membolehkan pentadbir sistem mengemaskini sistem mengikut keperluan sistem dalam masa yang tertentu.

Oleh itu, Modul Pentadbir ini membolehkan para pengguna sentiasa dapat menggunakan sistem yang sentiasa dikemaskinikan. Selain itu, segala kelemahan bagi sistem akan sentiasa diperbaiki dari masa ke masa dan prestasi sistem akan menjadi lebih baik.

4.2 Keperluan Bukan Fungsian

Suatu keperluan bukan fungsian atau kekangan menghuraikan tentang batas sistem yang menghadkan pilihan untuk membina suatu penyelesaian kepada permasalahan. Selain itu ia juga adalah keperluan yang sepatutnya diperolehi dalam sesebuah sistem bagi memastikan sistem yang dibangunkan itu mencapai tahap yang tinggi dan berkesan.

i. Antaramuka yang Ramah Pengguna

Antaramuka yang direkacipta mestilah mempunyai ciri-ciri yang ramah pengguna, mudah difahami dan digunakan. Metafor antaramuka yang digunakan patut

membolehkan para pengguna menghubungkan satu paparan ke paparan seterusnya dengan cepat dan berkesan melalui rekacipta yang konsisten dan grafikal. Secara amnya rekabentuk sistem ini memenuhi kriteria-kriteria berikut:

- ❖ Konsisten dari segi rekabentuk skrin dan paparan mesej ralat.
- ❖ Pemahaman yang mudah kerana semua prosedur tidak memerlukan kata arahan ataupun hafalan.

ii. Kebolehselenggaraan dan bermodular

Semua keperluan yang dipenuhi oleh EMVS adalah dibangunkan secara bermodular agar ia senang diselenggara. Pendekatan modular di dalam pengekodan program ini adalah satu proses memecahkan satu rutin kompleks dipecahkan kepada beberapa modul kecil. Ia digunakan agar kod program mudah dibaca dan diselenggara.

iii. Rekabentuk dan Kestabilan Paparan

Penekanan terhadap beberapa aspek perlu diambil kira dalam persembahan sesebuah paparan dan juga kesannya. Antara penekanan tersebut adalah:

- ❖ Kualiti dan keringkasan teks juga perlu dititik beratkan. Teks yang akan digunakan dalam sistem ini adalah yang mudah dibaca oleh pengguna. Keringkasan teks juga penting kerana penerangan yang panjang lebar tidak diperlukan pada paparan kerana ia mengambil masa untuk difahami oleh pengguna. Penerangan yang ringkas dan padat serta mudah difahami oleh pengguna adalah amat digalakkan.
- ❖ Warna yang dipilih untuk merekabentuk latar belakang paparan juga perlu di ambil kira kerana ia memberi kesan langsung kepada pengguna. Jika

penggunaan warna yang kontra di antara satu sama lain di dalam satu paparan yang sama, menyebabkan pengguna berasa bosan kerana ia menyakitkan mata. Oleh itu, pemilihan warna yang sesuai adalah berdasarkan kepada warna yang memberi keselesaan kepada pengguna seperti warna-warna yang lembut.

- ❖ Kestabilan dalam rekabentuk suatu paparan bermakna ia memastikan setiap elemen interaktif berfungsi dengan betul sama ada pada kali pertama paparan tersebut digunakan atau pada masa-masa yang akan datang.

iv. Kebolehharapan

Sistem mestilah boleh diharap iaitu boleh mengelak daripada sistem terhenti ('system down'). Satu sistem diharapkan mempunyai kebolehharapan sekiranya ia tidak menghasilkan kesalahan program yang kritikal apabila digunakan dengan cara yang betul, iaitu cara yang sepatutnya digunakan oleh pengguna biasa.

v. Keselamatan

Sistem ini mempunyai ciri-ciri keselamatan iaitu setiap pengguna yang ingin menggunakan sistem ini perlu melengkapkan bahagian login sebelum boleh memasuki sistem. Selain itu setiap maklumat dan data di dalam setiap modul tidak boleh diubahsuai oleh sesiapa kecuali pentadbir yang bertanggungjawab sahaja.

vi. Masa Maklumbalas

Masa maklumbalas hendaklah berada dalam lingkungan masa yang munasabah di dalam menjalankan sesuatu aktiviti dalam sistem ini. Maklumat yang dipaparkan juga haruslah ringkas dan padat bagi mengelakkan sebarang proses operasi yang terlalu lama.

43 Keperluan Perisian

i. Microsoft Visual Basic 6.0

Setelah menilai kemampuan sistem dan fungsi-fungsi yang kompleks, pemilihan terhadap perisian pembahagian sistem yang baik adalah perlu. Untuk itu, perisian pengaturcaraan utama yang digunakan untuk membangunkan Email Virus Scanner (EMVS) ini ialah Microsoft Visual Basic 6.0. Microsoft Visual Basic 6.0 adalah berasaskan antaramuka pengguna bergrafik dan bersifat 'even driven' di mana sesuatu objek boleh dibina dengan mudah.

Setiap fungsi yang dilakukan oleh objek akan dikodkan dengan cepat kerana penekanan hanya diberikan kepada fungsi yang akan dilakukan oleh objek tersebut. Tambahan pula untuk merekabentuk, objek dapat dikurangkan dengan menggunakan bahasa pengaturcaraan ini. Perisian ini digunakan kerana ia merupakan satu perisian yang sesuai untuk membangunkan EMVS yang bercirikan antaramuka pengguna yang sangat baik.

Di samping itu juga, ia membenarkan kawalan-kawalan seperti 'button', 'checkboxes', 'dropdownlistboxes' dan 'editboxes' yang memudahkan pembangunan dan penggunaan. Aplikasi perisian ini juga menyediakan kemudahan untuk ditukarkan kepada pemacu-

pemacu bagi kebanyakan sistem pangkalan data hubungan (RDBMS) yang ada seperti Oracle, sysbase, Informix, DB2, Microsoft SQL Server, Paradox dan lain-lain.

Perisian ini juga menyediakan kemudahan perpustakaan (DLL) yang baik dan fungsi-fungsi terbina dalam ('built in function'). Ia juga membenarkan pengaturcara mencipta perpustakaan sendiri dalam bahasa lain seperti bahasa C++ untuk dilarikan bersama perisian ini. Kebiasaan pengaturcara menyediakan perpustakaan mereka menggunakan bahasa C++ yang dikenali sebagai enjin. Tetapi bagi antaramuka ('interface'), pengaturcara menggunakan perisian Microsoft Visual Basic 6.0 kerana persembahannya yang sangat baik yang berkonsepkan Pengaturcaraan Berorientasikan Objek (OOP) yang mempunyai kelebihan dalam teknik kelas, perwarisan dan polimorfisme yang membolehkan tugas-tugas pembangunan digunakan semula ('reusable') dan dilanjutkan ('extensible') dengan lebih cekap.

Kelebihan Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 adalah salah satu peralatan 'Computer Aided Software Engineering' (CASE tool) yang popular masa kini. Peralatan CASE ini telah banyak membantu pembangunan sistem terutamanya dalam memudahkan kerja serta masa pembangunan.

Bagi mencapai tahap terbaik dalam fasa analisis, rekabentuk dan implementasi sistem maklumat, kita perlu menguruskannya supaya mencapai matlamat projek secara produktif. Dalam beberapa tahun yang lalu, penganalisis telah dimudahkan dengan

peralatan produktif baru yang telah dicipta untuk meningkatkan rutin kerja mereka bagi pembahagian sistem komputer. Peralatan CASE terbahagi kepada tiga iaitu :

- I. Lower CASE
- II. Upper CASE
- III. Integrasi CASE (gabungan Lower CASE dan Upper CASE)

❖ Lower CASE

Peralatan Lower CASE digunakan untuk menjanakan sumber kod komputer iaitu mengelak keperluan untuk pengaturcaraan program. Misalnya peralatan CASE Teleuse yang dibangunkan oleh Telesoft, menjanakan kod program dalam persekitaran motif.

❖ Upper CASE

Peralatan Upper CASE membolehkan pembangun mencipta dan mengubahsuai rekabentuk system. Misalnya SUIT yang dibangunkan oleh Randy Pausch di Universiti Virginia sebagai pembina antaramuka dan supercard yang dibangunkan oleh Aldus dalam persekitaran Macintosh & DOS yang merupakan satu pemprototaip.

❖ Integrasi CASE

Peralatan Integrasi CASE pula merupakan gabungan kedua-dua jenis CASE itu di mana kedua-dua cara implementasi peralatan digunakan seperti yang telah dirancangkan. Di antara contoh-contoh alatan CASE ialah :

- I. Visual Basic yang dibangunkan oleh Microsoft Corp

II. Power Builder yang dibangunkan oleh Microsoft Corp

III. Smaltalk IV yang dibangunkan oleh Digital Inc

IV. Delphi yang dibangunkan oleh International Inc

Terdapat peralatan CASE yang dikhususkan kepada fungsi-fungsi tertentu seperti berikut :

I. Sistem pembahagian aplikasi

II. Pemprototaip

III. Sistem pengurusan antaramuka pengguna

IV. Penjanaan kod

V. Peralatan pakej

Bagi membangunkan EMVS, peralatan CASE yang digunakan adalah Visual Basic versi 6.0, edisi 'enterprise'. Tujuan peralatan ini digunakan adalah seperti berikut :

I. Berfungsi sebagai sistem pembahagian aplikasi dan pemprototaipan

II. Jenis peralatan integrasi Windows

III. Meningkatkan produktiviti pembangunan

IV. Memajukan komunikasi para analisis dengan pengguna

4.4 Keperluan Sistem Pengoperasian

Di dalam membangunkan EMVS ini, Windows NT/2000 dipilih sebagai sistem pengoperasian untuk komputer peribadi. Ciri-ciri bagi sistem pengoperasian Windows NT/2000 adalah seperti berikut :

i. Windows NT/2000

NT seperti yang tertera dibelakang perkataan Windows adalah bermaksud Teknologi Baru manakala Windows 2000 pula adalah peningkatan teknologi yang berasaskan Windows NT. Bahagian antaramuka ini adalah yang paling jelas, dengan skrin yang berwarna-warni dan kadangkala mempunyai ikon yang kecil. Windows NT/2000 juga menawarkan faedah yang banyak seperti sistem operasi yang lain. Dengan menggunakan faedah-faedah ini, pembangun boleh menjadi produktif dan menerbitkan kandungan sistem mereka ke Internet dengan menggunakan kebolehan Windows NT/2000. Berikut adalah ciri-ciri Windows NT/2000 :

❖ Ramah pengguna dan senang digunakan

Windows NT/2000 senang digunakan kerana ikon yang terdapat pada antaramukanya adalah tidak mengelirukan pengguna.

❖ Modul Ingatan 32-bit

Windows NT/2000 adalah sistem operasi 32-bit yang menggunakan 32-bit alamat untuk menerusi sesuatu objek. Banyak kelebihan dapat dilihat. Di antaranya ialah ia membolehkan teknologi baru di alamatkan iaitu 4,19,304KB (four gigabytes) ingatan.

❖ Tiada lagi DOS

Walaupun tiada lagi DOS, Windows NT masih boleh melarikan program utama DOS selagi pembangun sistem tidak terus masuk ke dalam perkakasan atau memerlukan 'driver' yang khas. Ini adalah dengan mewujudkan satu persekitaran DOS yang maya iaitu dipanggil NTVDM (NT Virtual DOS Machine). Program DOS dilarikan di dalam persekitaran yang telah wujud ini.

❖ **Sistem Operasi Rangkaian**

Windows NT adalah berfungsi untuk sistem operasi yang biasa dan juga sistem operasi rangkaian. Dengan pengurus LAN, OS/2 adalah sistem operasi dan pengurus LAN bertindak sebagai sistem operasi rangkaian. Integrasi ini dengan OS dan DOS telah membuktikan bahawa terdapatnya kombinasi yang sesuai di dalam Windows NT/2000.

❖ **Kebolehharian Terhadap Model Ingatan**

Di dalam Model Ingatan Windows NT, segala proses mendapat 32-bit ruang alamat. Ruang 4GB ini dibahagikan kepada setengah iaitu aplikasi hanya menggunakan 2GB ruang. Selebihnya adalah digunakan untuk antaramuka dan sebagainya. Setiap proses akan berfikir dan membuat segala benda secara efektif. Tiada cara lain ia akan baca atau tulis di luar ruang ingatan samada secara sengaja atau tidak sengaja. Ini boleh mengelakkan sistem daripada rosak atau 'crash' dan ia menyediakan ciri-ciri keselamatan yang dikehendaki pengguna.

❖ **Persendirian / Kesesuaian**

Windows NT/2000 direkabentuk untuk menyokong pelbagai personaliti. Antaramukanya menjadi personaliti primary. Ia juga menyokong personaliti POSIX, personaliti OS2 dan juga personaliti DOS/Windows. Tambahan personaliti, seperti UNIX juga boleh disokong.

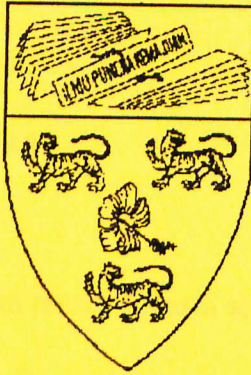
❖ **Keselamatan**

Windows NT/2000 direka khusus untuk memenuhi peringkat keselamatan negara Amerika Syarikat iaitu Agency's CA Level. Dengan merekacipta Windows NT

menggunakan model keselamatan ini, Microsoft boleh meyakinkan pembeli perisian yang mana perisian ini mempunyai ciri-ciri keselamatan yang unggul.

❖ **Yuran Lesen**

Walaupun, Windows NT/2000 mempunyai perisian hakcipta terpelihara yang memerlukan yuran lesen untuk setiap sumber kod yang digunakan



BAB 5 BENTUK SISTEM

BAB 5 REKABENTUK SISTEM

5.1 Rekabentuk Sistem

Rekabentuk sistem adalah suatu proses di mana segala keperluan akan ditafsirkan dalam bentuk persembahan perisian. Persembahan perisian ini dikenali sebagai rekabentuk antaramuka komputer. Antaramuka pengguna sesuatu sistem selalunya merupakan ciri utama untuk mengenalpasti sama ada sesuatu sistem itu merupakan sistem yang baik atau tidak untuk digunakan. Antaramuka sistem yang sukar untuk digunakan akan mengakibatkan kekerapan pengguna membuat kesilapan. Keadaan ini akan menyebabkan sistem perisian itu tidak akan digunakan lagi oleh orang ramai kerana tidak puas hati dengan fungsi- fungsi yang terdapat pada sistem tersebut.

Walaupun antaramuka yang bercirikan teks sudah pasti akan kekal kegunaannya pada masa akan datang, kebanyakan pengguna mengharapkan aplikasi sistem mempunyai ciri- ciri antaramuka bergrafik yang mana lebih dikenali sebagai Antaramuka Pengguna Bergrafik (GUI).

5.2 Rekabentuk Antaramuka Grafik (GUI)

Antaramuka pengguna bergrafik mempunyai beberapa kelebihan jika dibandingkan dengan antaramuka yang biasa digunakan iaitu antaramuka pengguna berdasarkan teks, iaitu:

- I. Antaramuka pengguna bergrafik secara relatifnya menyenangkan penggunaan sesuatu sistem itu. Jadi, pengguna yang tidak mempunyai pengetahuan komputer boleh belajar untuk menggunakan antaramuka yang disediakan dengan mudah setelah mendapat penerangan yang ringkas tentang antaramuka tersebut.
- II. Antaramuka jenis ini juga dapat menyediakan banyak tettingkap (*windows*) kepada pengguna untuk berinteraksi dengan sistem. Jadi, apabila berlaku pertukaran daripada satu tugas ke tugas yang lain, hasil tugas yang dilakukan pada awalnya dapat dilihat juga pada skrin semasa melakukan tugas yang kedua dan seterusnya.
- III. Masa interaksi yang cepat dan juga berskrin penuh dapat dilakukan dengan akses cepat pada mana- mana bahagian pada skrin.

Antaramuka pengguna bergrafik juga mempunyai beberapa ciri yang membezakannya dengan antaramuka berasaskan teks, iaitu:

- I. '*Windows*' - menyediakan banyak skrin pada satu masa untuk membolehkan penerangan berbeza dipaparkan serentak pada skrin.

- II. Ikon - Ikon merujuk kepada elemen grafik yang memuatkan jenis maklumat yang berbeza. Dalam sistem, terdapat ikon yang merujuk kepada fail, merujuk kepada suatu proses dan sebagainya.
- III. Menu - Membolehkan pengguna memilih bahagian tertentu dalam sistem dengan hanya menggunakan penuding atau papan kekunci. Ia memudahkan pengguna berbanding dengan antaramuka berasaskan teks .
- IV. Penuding - Alatan penuding seperti tetikus digunakan untuk membuat pilihan daripada menu atau menandakan sesuatu yang diperlukan pada '*Windows*'.

5.3 Rekabentuk Pangkalan Data

Rekabentuk pangkalan data melibatkan penakrifan struktur pangkalan data di mana Sistem Pengurusan Pangkalan Data (DBMS) menyimpan fakta-fakta tentang struktur-struktur tersebut di dalam pangkalan data itu sendiri. Ia mengubah maklumat yang tidak berstruktur dan pemprosesan keperluan kepada suatu perwakilan yang menakrifkan spesifikasi-spesifikasi fungsian.

Semua data dan hubungan diwakilkan dalam jadual 2-dimensi yang terdiri daripada baris dan lajur. Setiap silangan baris dan lajur yang dikenali sebagai sel hanya boleh mempunyai satu nilai sahaja. Sistem Pengurusan Pangkalan Data Hubungan (RDBMS) menyembunyikan kekompleksan sistem dari pengguna/pembangun sistem.

Data-data yang diperolehi semasa sesi temubual telah digunakan untuk mengenalpasti atribut-atribut bagi jadual-jadual modul dan sub-modul sistem EMVS. Setiap jadual mempunyai satu atau gabungan kekunci primer yang mempunyai nilai unik.

Penormalan dilaksanakan semasa proses rekabentuk pangkalan data. Ia merupakan satu proses penyusunan elemen data ke dalam rekod-rekod. Tujuan utama penormalan adalah untuk mengurangkan jumlah data yang akan disimpan melalui pengurangan data-data yang bertindan. Terdapat tiga langkah utama di dalam proses penormalan iaitu :

1. Bentuk Normal Pertama (1NF)

Melibatkan proses penghapusan kumpulan-kumpulan berulang dan menghasilkan satu jadual baru di mana semua atribut kunci ditakrifkan dan saling bersandar terhadap kunci primer.

2. Bentuk Normal Kedua (2NF)

Merupakan proses pengasingan atribut-atribut yang bersandar hanya terhadap sebahagian daripada kunci primer. Ia lebih dikenali dengan penyingkiran kesandaran separa. Atribut-atribut yang berkenaan ini diletakkan di dalam jadual yang berasingan. Penormalan ini memastikan semua atribut bukan-kunci adalah bersandar sepenuhnya terhadap kunci primer.

3. Bentuk Normal Ketiga (3NF)

Proses ini melibatkan penghapusan sebarang persandaran transitif iaitu kesandaran yang wujud apabila atribut bukan-kunci bersandar terhadap atribut bukan kunci yang lain.

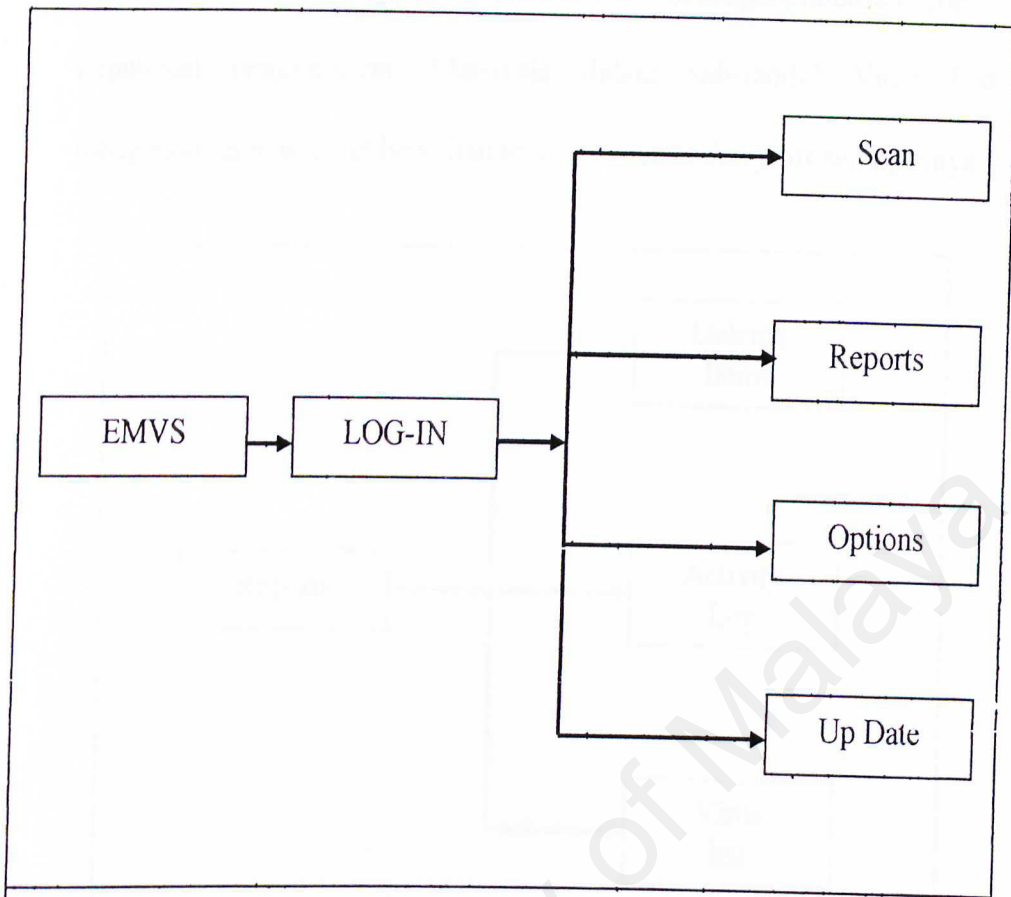
5.4 Rekabentuk EMVS

Rekabentuk program merupakan proses di mana keperluan-keperluan sistem diterjemahkan kepada perwakilan perisian. EMVS direkabentuk berdasarkan pendekatan pembangunan modular di mana sistem dibahagikan kepada modul-modul dan kesemua sub-modul. Rekabentuk dimulakan dengan sub-modul terbawah sekali.

EMVS merupakan satu sistem berpandu-menu ('menu driven') yang terdiri daripada beberapa paras menu pada skrin utama sistem. Ia terdiri daripada 3 modul utama iaitu:

- I. Scan
- II. Reports
- III. Options
- IV Up Date

Carta hierarki EMVS ditunjukkan pada rajah di bawah:



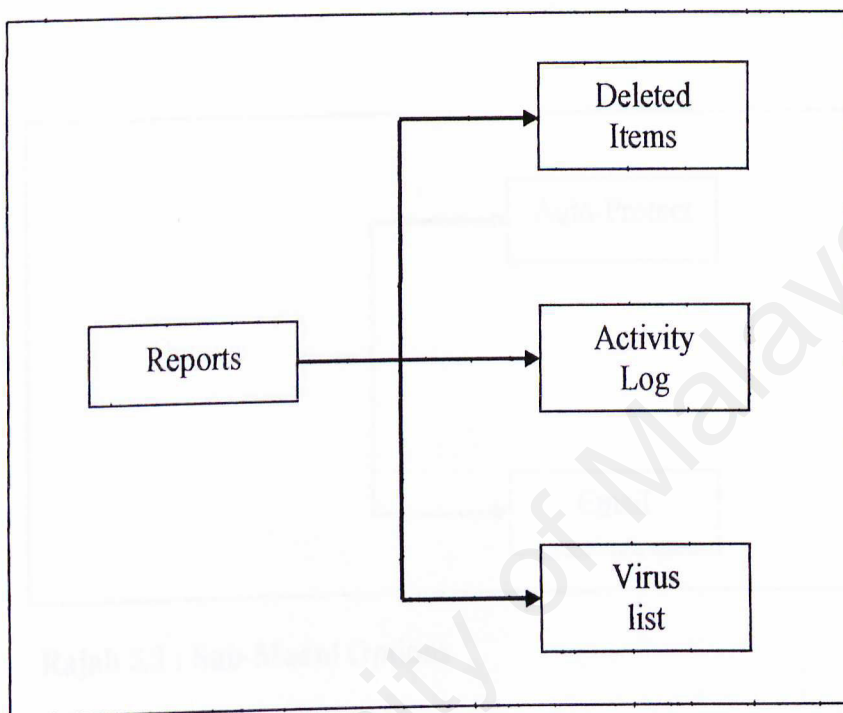
Rajah 5.1 : Carta Hierarki EMVS

5.4.1 Modul Reports

Modul Reports ini terbahagi kepada 3 sub-modul utama iaitu sub-modul **Deleted Items**, **Activity Log** dan **Virus List** seperti yang ditunjukkan pada rajah di bawah.

Dalam sub-modul Deleted Items, ia mengandungi maklumat berkenaan nama fail yang dihapuskan, lokasi asal serta tarikh dan masa fail tersebut dikuarantin. Ia juga mempunyai pilihan samada untuk memadam virus atau tidak. Dalam sub-modul Activity Log pula, ia mengandungi maklumat berkenaan masa dan tarikh

pengimbasan virus, status pengimbasan iaitu samaada pengimbasan dilaksanakan, dibatalkan ataupun telah selesai, lokasi pengimbasan serta keputusan pengimbasan. Manakala dalam sub-modul Virus List pula mengandungi maklumat berkaitan jenis-jenis virus dan jenis serangannya.



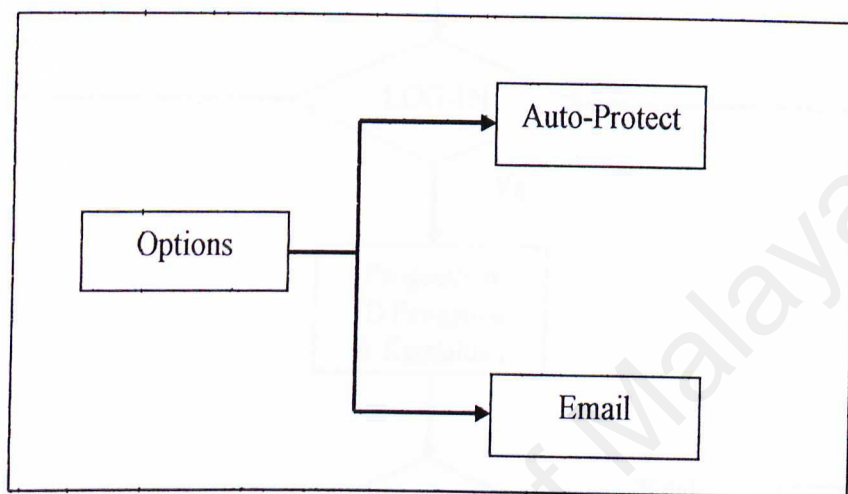
Rajah 5.2 : Sub-Modul Reports

5.4.2 Modul Options

Modul Options ini pula terbahagi kepada 2 sub-modul utama iaitu sub-modul **Auto-Protect** dan sub-modul **Email** seperti yang ditunjukkan pada rajah di bawah.

Dalam sub-modul Auto-Protect, ia mempunyai pilihan samaada hendak memulakan pengimbasan secara automatik ketika memasuki sistem operasi ataupun tidak. Dalam sub-modul Email pula, ia mengandungi pilihan samaada

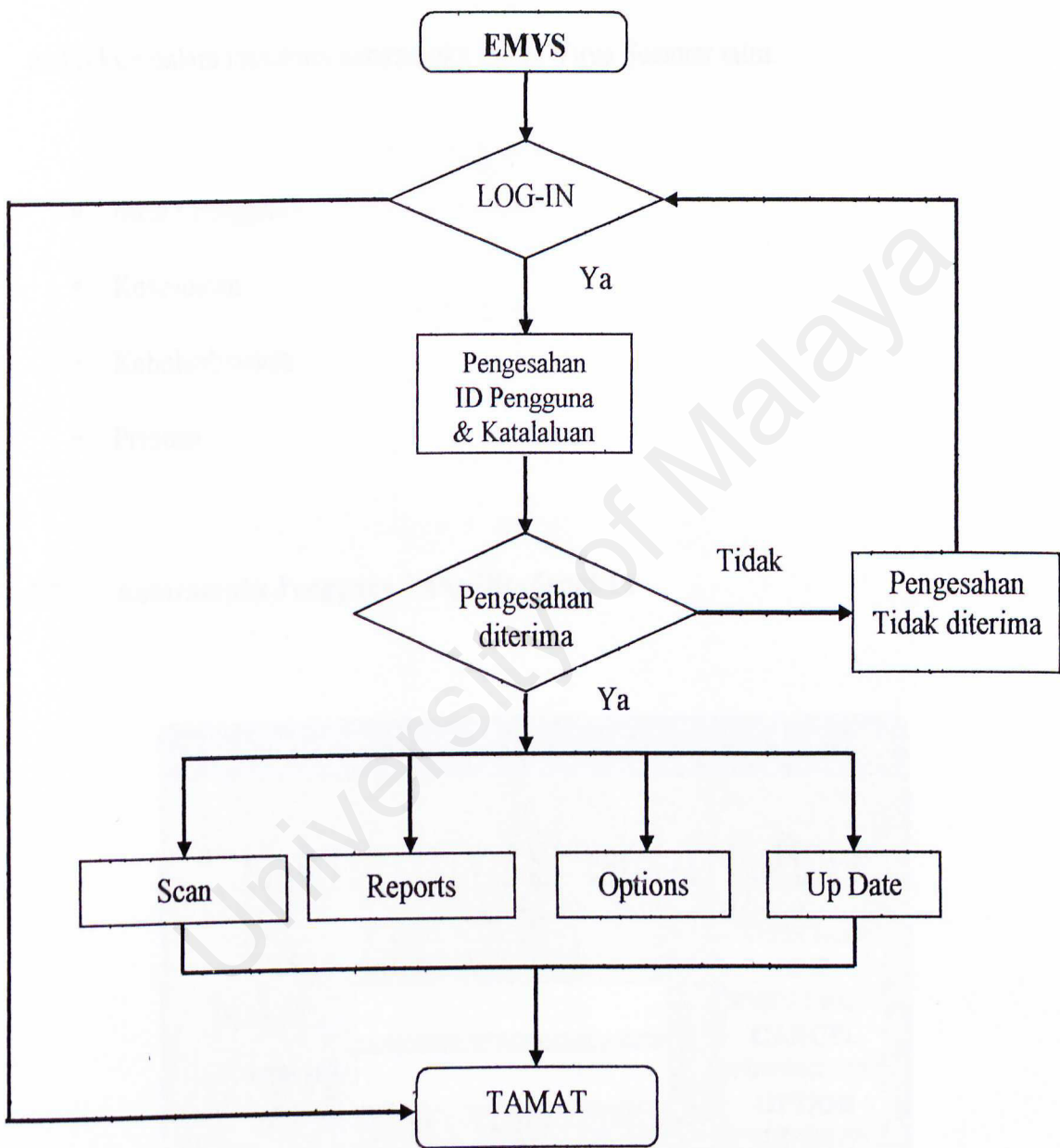
hendak terus mengimbas emel yang datang ('incoming email') atau tidak, pilihan samaada hendak terus menghapuskan fail yang diserang/dijangkiti ataupun tidak dan pilihan samaada hendak memberitahu pengguna secara automatik tentang serangan pada emel yang diterimanya ataupun tidak.



Rajah 5.3 : Sub-Modul Options

5.5 Carta Aliran Sistem

Carta aliran ini adalah untuk menunjukkan tentang capaian pengguna terhadap sistem ini. Carta aliran bagi EMVS ditunjukkan pada rajah di bawah.



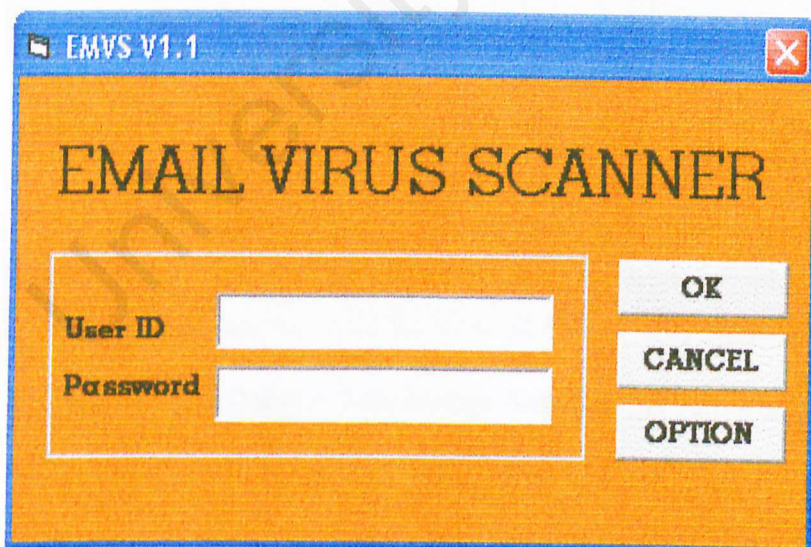
Rajah 5.4 : Carta Aliran Bagi EMVS

5.6 Rekabentuk Antaramuka Pengguna

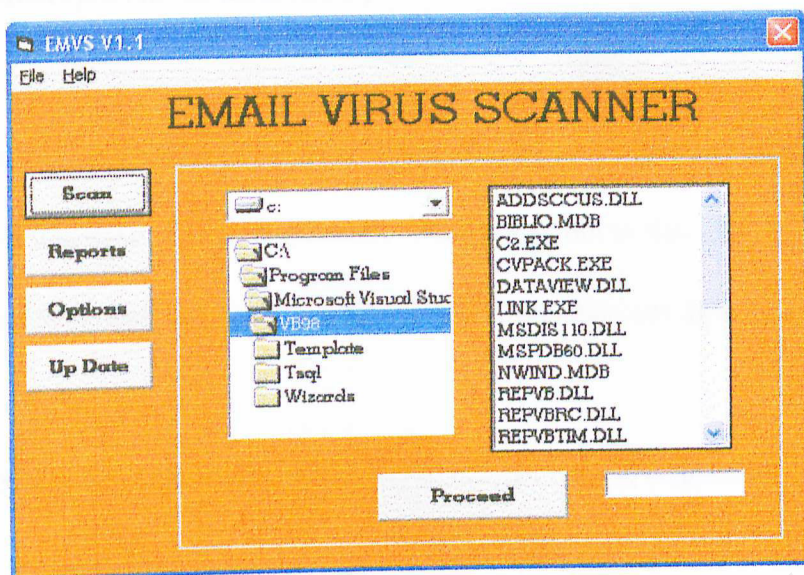
Agar Email Virus Scanner (EMVS) ini dapat diterima dan digunakan dengan efektif, antaramuka perlu direkabentuk untuk kemudahan pengguna. Beberapa faktor telah diambil kira dalam membina antaramuka Email Virus Scanner iaitu:

- Mesra Pengguna
- Kesesuaian
- Kebolehgunaan
- Prestasi

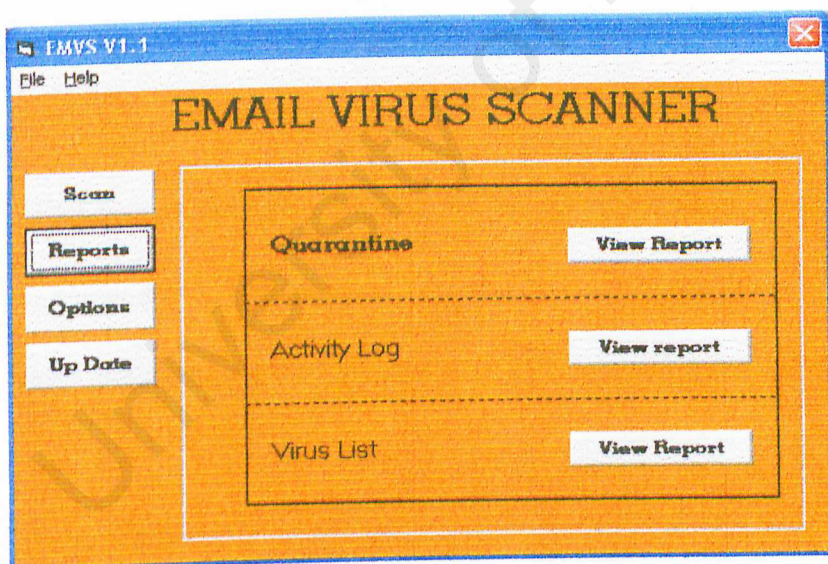
5.7 Antaramuka Pengguna Yang Dicadangkan



Rajah 5.5 : Cadangan Rekabentuk Antaramuka log-in



Rajah 5.6 : Cadangan Rekabentuk Antaramuka Scan



Rajah 5.7 : Cadangan Rekabentuk Antaramuka Reports

5.8 Gambarajah Aliran Data (DFD)

Salah satu teknik utama pemodelan data, dalam mengadakan keperluan maklumat adalah gambarajah aliran data (data flow diagram). Gambarajah aliran data (DFD) secara grafik menawan dan memodelkan bagaimana data mengalir dan diproses dalam sistem, iaitu :

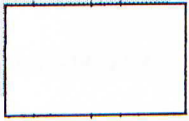
- ❖ Secara konseptual memodelkan (logikal dan fizikal) bagaimana data bergerak dalam sistem.
- ❖ Menerangkan proses (penukaran) yang dilalui oleh data.
- ❖ Menakrif input dan output bagi sistem.

Kelebihan utama menggunakan gambarajah aliran data ialah:

- ❖ Kebebasan untuk memodelkan sistem secara berkonsep tanpa dibatasi oleh teknologi dan implementasi.
- ❖ Lebih memahami kerumitan sistem dan hubungan subsistem-subsistem yang ada.
- ❖ Memudahkan serta membolehkan komunikasi (bergrafik) bagi spesifikasi sistem ke pengguna akhir.
- ❖ Memastikan penganalisis mengenalpasti kesemua input, output dan proses.

Gambarajah aliran data mempunyai simbol-simbol tertentu yang mempunyai maksud tersendiri. Antara simbol asas DFD yang telah digunakan semasa membangunkan EMVS ini ialah:

I. Entiti luaran



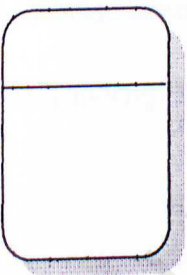
- Sebarang entiti yang boleh menghantar dan menerima data sistem, punca atau destinasi.
- Dianggap di luar sempadan sistem.

II. Aliran



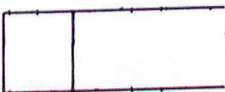
- Menunjukkan pergerakan data dari satu titik ke suatu titik (anak panah menghala ke destinasi)
- Jika aliran data berlaku serentak, maka ia digambarkan sebagai anak panah selari menghala ke arah yang berlawanan.

III. Proses



- Menunjukkan berlakunya proses penukaran data.
- Nama aliran data yang memasuki proses adalah berlainan dari yang meninggalkan proses.
- Nomborkan proses dengan nombor yang unik.

IV. Storan data



- Mewakili storan data kekal atau separa kekal – fail digital atau pangkalan data.
- Nomborkan storan data dengan nombor unik.

Gambarajah aliran data (DFD) boleh dan sepatutnya dilukis secara sistematik. Dalam projek ini, pendekatan atas-bawah telah digunakan. Gambarajah aliran data (DFD) bergerak dari yang sifat umum menjurus ke arah yang lebih spesifik. Berikut adalah langkah-langkah yang digunakan dalam mengadakan gambarajah aliran data (DFD):

I. Gambarajah konteks

II. Gambarajah 0

5.8.1 Gambarajah Konteks

Gambarajah konteks merupakan tahap tertinggi di dalam gambarajah aliran data (DFD) dan ia mengandungi hanya satu proses sahaja. Proses ini diberi nombor unik 0. Ia dibangunkan dari proses pengumpulan maklumat. Pada gambarajah konteks ini, kesemua entiti luaran dan aliran data utama ditunjukkan. Gambarajah tidak mengandungi storan data.

Pembangunan EMVS secara keseluruhannya boleh dimodelkan dengan menggunakan Gambarajah Konteks yang terdapat pada Apendiks C. Gambarajah Konteks tersebut menunjukkan perhubungan di antara pentadbir dan sistem.

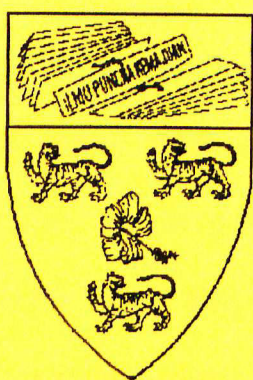
5.8.2 Gambarajah 0

Gambarajah 0 merupakan tahap kedua di dalam gambarajah aliran data. Ia merupakan ledakkan proses gambarajah konteks menjadi sub-proses dan lebih

menunjukkan perincian setiap proses. Ia tidak sepatutnya mempunyai lebih daripada 9 proses dan setiap proses dinomborkan dengan nombor bulat. Storan data juga ditunjukkan pada gambarajah 0. Ia dinomborkan dengan nombor bulat yang dimulai dengan huruf D.

Gambarajah 0 bagi EMVS boleh dirujuk pada Apendiks D. Ia terdiri daripada 3 proses utama iaitu Pengesahan, Reports dan Options. Selain itu, ia juga merangkumi tiga pangkalan data iaitu :

- I. Fail ID Pentadbir yang menyimpan nama pengguna dan kata laluan.
- II. Fail Maklumat Reports yang menyimpan maklumat berkaitan dengan Reports iaitu Quarantined Items, Activity Log dan Virus List
- III. Fail Maklumat Options yang menyimpan maklumat berkaitan dengan Options iaitu Auto-Protect dan Email.



BAB 6

PELAKSANAAN SISTEM

6.1 Pengenalan

Setelah fasa rekabentuk selesai, maka fasa seterusnya dimulakan iaitu Fasa Perlaksanaan Sistem. Dalam fasa ini pengkodan dilakukan dan semua media digabungkan untuk menjadi sebuah program. Di dalam pembangunan EMVS, pengkodan telah dilakukan serentak dengan pembinaan antaramuka sistem.. Dalam fasa ini, saya telah cuba mempertimbangkan perkara-perkara berikut :

- I. Faktor ketahanan, iaitu sejauh mana fungsi-fungsi yang dihasilkan dapat bertindak terhadap data-data yang digunakan oleh pengguna. Contohnya, adakah sistem akan tergantung (*hang*) apabila input data adalah di luar julat.
- II. Pengkodan yang mudah difahami, iaitu kod program yang dibina mudah dibaca dan difahami kerana menyediakan komen bagi setiap fungsi yang dibina.
- III. Mudah diselenggarakan, iaitu membolehkan sistem dikemaskini dengan mudah. Penyelenggaraan sistem ini boleh dilakukan oleh pengguna yang mempunyai kebenaran untuk mengubahsuai data-data tertentu secara terus dari sistem.

6.2 Proses – Proses Yang Terlibat

Antara proses-proses yang terlibat dalam fasa ini ialah :

1 Paparan teks

Paparan teks bagi setiap modul dipilih dan diubahsuai dengan menggunakan teks editor (*prompt properties*). Pemilihan warna, jenis dan saiz huruf dibuat berdasarkan kesesuaian dengan paparan modul terbabit.

II Pautan antara modul-modul

Proses ini membolehkan input yang baru diterima oleh suatu modul dihantar ke modul yang lain tanpa berlaku sebarang ralat. Contoh seperti borang Log In yang boleh dihubungkan kepada borang utama apabila pengguna telah memasukkan nama pengguna dan kata laluan yang betul. Berikut adalah contoh kod tersebut :

```
Private Sub Command3_Click()
```

```
If txtLogin.Text = "login" And txtPassword.Text = "password" Then
```

```
Form1.Show
```

```
Unload Me
```

```
End If
```

```
End Sub
```

```
Private Sub
```

```
Command6_Click()
```


End

End Sub

Private Sub Text1_Change()

End Sub

Private Sub txtPassword_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then Command3_Click

End Sub

Private Sub txtLogin_KeyPress(KeyAscii As Integer)

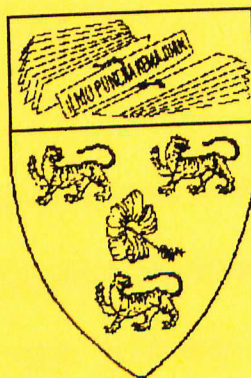
If KeyAscii = 13 Then Command3_Click

End Sub

6.3 Pengaturcaraan

Pengaturcaraan atau pengkodan adalah antara proses yang penting di dalam membangunkan suatu sistem. Ia akan menentukan sama ada suatu sistem itu berjaya mencapai objektifnya. Oleh itu persekitaran pembangunannya adalah penting bagi menjamin proses pengaturcaraan ini dapat dilakukan dengan teliti dan seterusnya menghasilkan sistem yang baik. Terdapat tiga aspek utama yang dititik beratkan di dalam proses pengkodan ini iaitu *Struktur Kawalan*, *Algoritma* dan *Struktur Data*.

Dalam fasa pelaksanaan ini, langkah penyusunan kesemua proses-proses yang diterangkan pada bahagian 5.2 dilakukan di mana proses tersebut diletakkan dalam 'cast' yang bersesuaian dengan kepenggunaannya dan keperluan sistem itu sendiri. Paparan proses ini dipilih berdasarkan ciri-ciri yang membolehkan sistem ini lebih jimat penggunaan ruang ingatan dan dapat dilarikan (*run*) dengan baik dan cekap.



BAB 7

KEKUALIFAN SISTEM

BAB 7 : PENGUJIAN SISTEM

7.1 Pengenalan

Pengujian merupakan satu proses untuk menguji keberkesanan sesuatu aturcara itu menjalankan fungsinya. Pengujian sistem adalah proses yang interatif. Pengujian sistem akan melibatkan proses pengesahan dan pentahkikkan terhadap sistem untuk memastikan kualiti sistem adalah seperti mana yang dikehendaki. Pengesahan merujuk kepada aktiviti untuk memastikan produk yang dijangka diperolehi, manakala pentahkikkan merujuk kepada aktiviti yang memastikan program dibangun sebagaimana spesifikasi.

Proses pengujian merupakan elemen yang paling penting bagi memastikan sama ada sistem yang dihasilkan memenuhi kehendak pengguna atau tidak. Sistem yang berkualiti mampu menjalani apa juga pengujian yang diberikan. Dengan itu, spesifikasi, rekabentuk dan aturcara yang telah dilakukan sepanjang proses pembangunan sistem akan dapat dibuat penelitian dan penilaian semula. Beberapa peraturan digunakan di dalam pengujian adalah berdasarkan kepada beberapa objektif iaitu :

- I. Pengujian adalah proses pelaksanaan program bertujuan mencari ralat yang mungkin wujud.
- II. Pengujian yang baik adalah pengujian yang dapat mengesan ralat yang belum di jumpai.

- III. Kejayaan sesuatu pengujian yang dapat mengesan ralat yang belum dijumpai

7.2 Jenis - Jenis Ralat

Semasa proses pengujian dijalankan, biasanya sistem akan memaparkan bentuk-bentuk ralat yang dilakukan oleh pengaturcara. Oleh itu adalah penting bagi pengaturcara untuk mengetahui jenis-jenis ralat ini. Semasa proses pengujian ini dilakukan, beberapa jenis ralat akan dapat dikesan iaitu seperti :

- I. **Ralat Algoritma**
- II. **Ralat Kompil**
- III. **Ralat Larian**
- IV. **Ralat Logik**
- V. **Ralat Dokumentasi**

7.2.1 Ralat Algoritma

Ralat algoritma berlaku apabila aturcara logik tidak menghasilkan keputusan yang diinginkan kerana ada ralat dalam langkah-langkah aturcara terutamanya dalam gegelung. Kesilapan ini mudah dikesan dengan menggunakan kaedah *test debug* iaitu dengan membaca langkah-langkah di dalam aturcara tersebut.

7.2.2 Ralat Kompil

Ralat kompil adalah ralat yang dihasilkan dari binaan kod yang salah.

Ralat kompil boleh dikesan semasa proses kompilan bagi pengkodan yang dibuat akibat kesilapan sinteks. Ralat ini dikesan secara terus dan boleh dibaiki segera.

7.2.3 Ralat Larian

Ralat larian wujud apabila pernyataan cuba untuk melakukan operasi yang tidak wujud. Contohnya menukar/mencipta objek yang tidak wujud.

7.2.4 Ralat Logik

Ralat logik wujud apabila program tidak melakukan fungsi tertentu sebagaimana yang dikehendaki. Ianya dikesan melalui produk yang dihasilkan oleh program. Ralat ini kadangkala sukar dikesan.

7.2.5 Ralat Dokumentasi

Ralat dokumentasi pula terhasil apabila dokumen tersebut tidak setara dengan hasil aplikasi. Kebiasaannya dokumentasi diperolehi daripada proses rekabentuk sistem dan menyediakan penerangan yang terperinci tentang apa yang aturcara tersebut lakukan. Tetapi semasa proses

pelaksanaan dijalankan, aturcara ini menghasilkan hasil yang sebaliknya.

Kesilapan ini akan mengakibatkan kesilapan-kesilapan yang seterusnya.

7.3 Jenis - Jenis Pengujian

Pengujian dilakukan dengan beberapa peringkat seperti berikut :

7.3.1 Pengujian Unit

Langkah pertama dalam pengujian adalah pengujian unit. Pengujian unit ini merangkumi pengujian ke atas setiap komponen modul aturcara itu sendiri dan diasingkan dari modul-modul yang lain dalam aplikasi. Setiap fail dalam modul yang sama akan berinteraksi antara satu sama lain dan ia juga akan berinteraksi dengan fail pada modul yang lain. Langkah-langkah di muka surat sebelah menerangkan bagaimana ujian unit dibuat ke atas sistem ini.

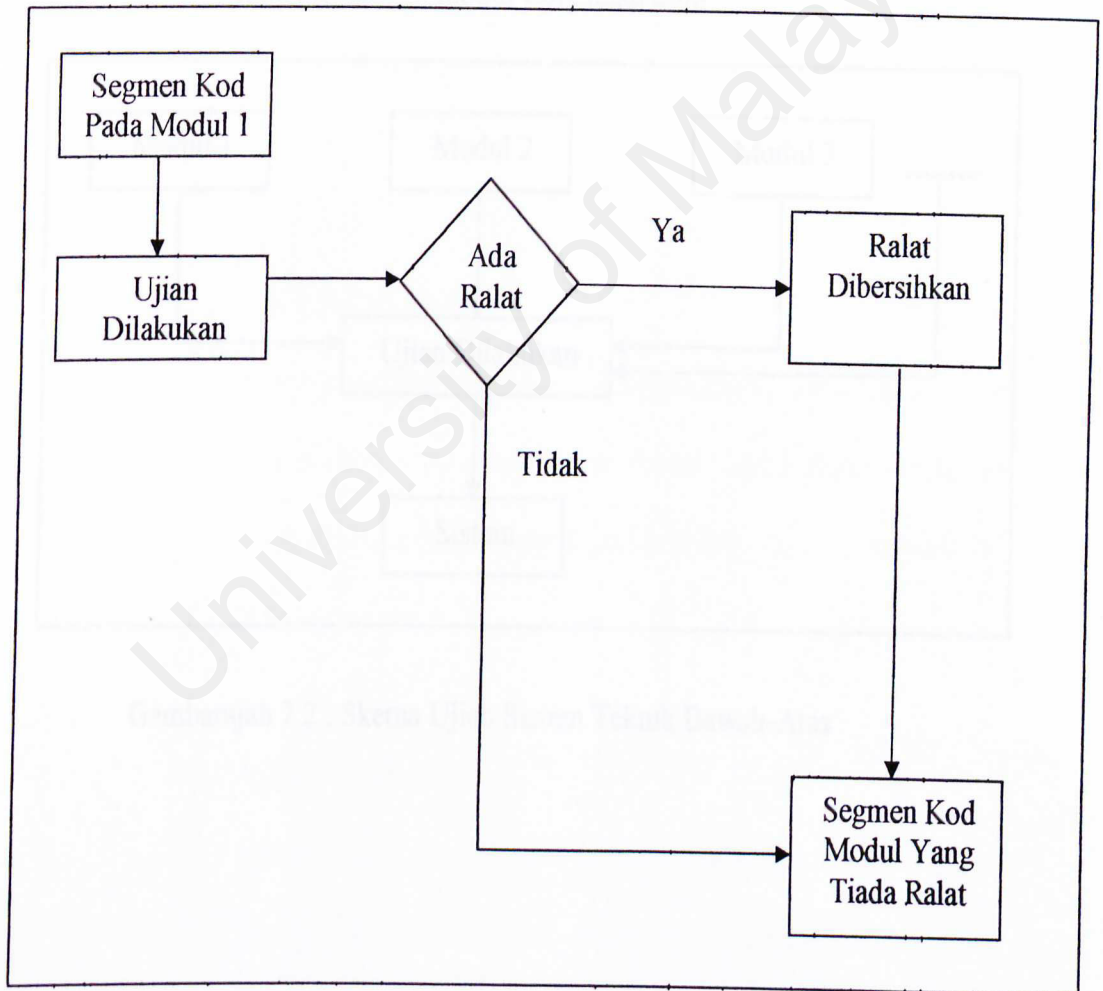
- I. Modul-modul diasingkan mengikut fungsi dan unit masing-masing
- II. Setiap kod di dalam program akan diperiksa dengan melihat dan membaca aliran kod aturcara untuk mengecam kesilapan algoritma dan kesilapan sinteks.

III. Kod aturcara dilarikan (*run*) menggunakan perisian Clarion 5 untuk mengenalpasti baki kesilapan yang tidak dikesan semasa proses penulisan kod tersebut.

7.3.2 Pengujian Modul

Modul adalah kombinasi beberapa unit berfungsi. Ujian dilakukan terhadap setiap modul yang ditakrif di dalam fasa rekabentuk. Ujian dilakukan untuk melihat aliran peristiwa adalah betul dan juga ralat logik.

Sila rujuk Gambarajah 7.1 : Skema Ujian Modul.



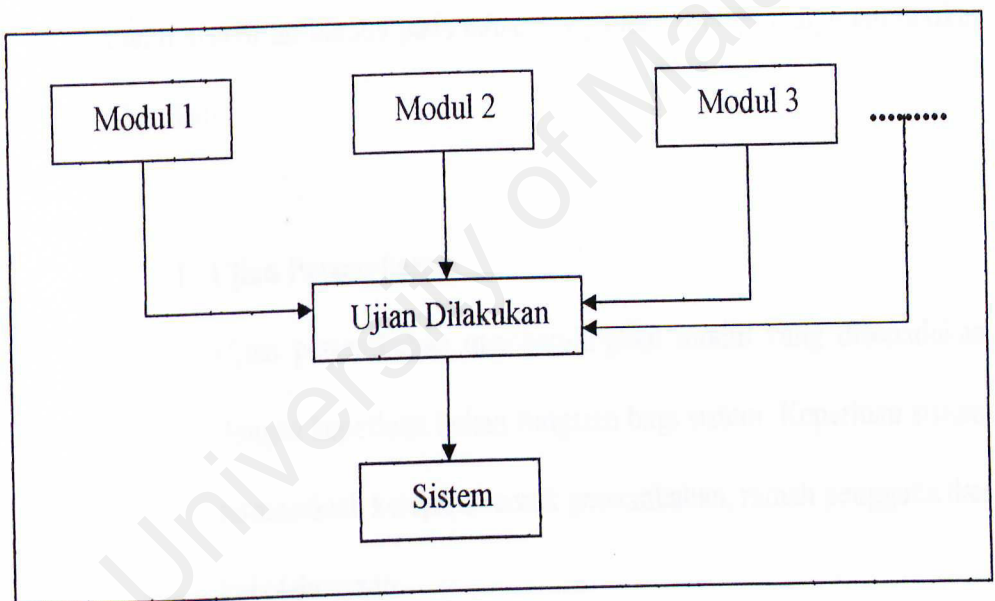
Gambarajah 7.1 : Skema Ujian Modul

7.3.3 Pengujian Integrasi

Ujian integrasi dilakukan untuk memastikan antaramuka di antara modul berjalan dengan baik. Pendekatan berikut digunakan di dalam ujian integrasi :

I. Pendekatan Bawah-Atas

Dalam pendekatan ini, modul peringkat bawah seperti rutin yang selalu dipanggil oleh modul lain di uji secara individu. Selepas itu, modul seterusnya yang akan diuji adalah modul yang memanggil modul yang telah diuji.



Gambarajah 7.2 : Skema Ujian Sistem Teknik Bawah-Atas

II. Pendekatan Atas-Bawah

Modul utama yang mengawal modul-modul lain diuji terlebih dahulu. Modul-modul yang akan di panggil oleh ini digabungkan dan diuji sebagai unit besar.

7.3.4 Pengujian Sistem

Ujian ini dijalankan setelah keseluruhan sistem telah siap. Pengujian dilakukan untuk memastikan fungsian dan semua elemen yang digabungkan sebagai satu sistem berfungsi dengan baik. Pengujian juga dilakukan untuk memastikan objektif sistem tercapai. Bagi memastikan kualiti sistem ini berada pada tahap yang baik, strategi pengujian berikut dilakukan.

I. Ujian Persembahan

Ujian persembahan membandingkan modul yang disepadukan dengan keperluan bukan fungsian bagi sistem. Keperluan sistem termasuklah ketepatan corak persembahan, ramah pengguna dan kebolehpercayaan.

II. Ujian Penerimaan

Ujian ini perlu menyemak ciri-ciri sistem untuk memastikannya memenuhi keperluan yang ditakrifkan. Kaedah yang di gunakan dalam ujian ini ialah meminta pengguna mencuba sistem yang telah siap dibina dan meminta pandangan

dan pendapat mereka supaya dapat dijadikan sebagai satu sumber maklumat bagi meningkatkan lagi keberkesanan pembangunan sistem di masa akan datang.

III. Ujian '*Volume*'

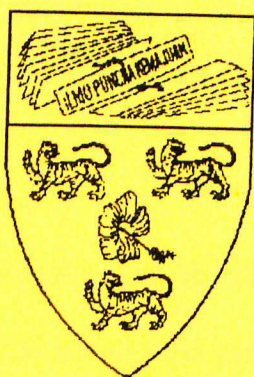
Ujian tahap medan dan rekod diperiksa sama ada ia boleh menerima segala kemungkinan data daripada pengguna.

IV. Ujian Keselamatan

Ujian ini adalah untuk memastikan bahawa aplikasi sistem yang dihasilkan memenuhi keperluan keselamatan seperti sukar dicerobohi.

V. Ujian Masa

Ujian ini dijalankan bagi memastikan prestasi persembahan sistem dari segi masa tindakbalas dan masa larian (*run*).



BAB 8

PERBINCANGAN

University of Malaya

BAB 8 PERBINCANGAN

8.1 Pengenalan

Di dalam bab ini, penilaian akan dibuat berdasarkan kepada kekuatan system, kekangan, keselamatan, masalah dan penyelesaian serta kesimpulan daripada system ini.

8.2 Kekuatan Sistem

- I Sistem ini mudah untuk dibawa ke mana-mana kerana saiznya yang kecil. Pengguna terutamanya pelajar yang biasa menggunakan komputer di makmal komputer ataupun di pusat internet boleh membawanya di dalam thumbdrive dan mengaktifkannya dengan hanya menekan satu icon sahaja dan dapat mengimbas serta megesan beberapa jenis virus. Sistem ini mengurangkan kebergantungan terhadap antivirus sedia ada.

- II Sistem ini hanya membenarkan pengguna yang berdaftar sahaja untuk mencapai sebarang maklumat. Setiap kali sistem diaktifkan maka pengguna perlu untuk log in semula walaupun sistem telah dilarikan sebelumnya. Keselamatan sistem merupakan factor yang penting dalam menjaga kebolehpercayaan maklumat.

- III antaramuka yang ramah pengguna bagi membolehkan pengguna mengendalikannya dengan mudah. Butang-butang arahan yang terdapat pada antaramuka sistem ini senang difahami dan digunakan.
- IV Sistem ini menyediakan mesej jika pengguna melakukan kesilapan semasa menggunakan sistem ini.
- V EMVS v1.1 ini membenarkan pengguna untuk mengupdate virus baru. Pengguna hanya perlukan virus signature untuk ditambahkan kedalam sistem ini. Jadi antivirus yang terdapat dalam sistem ini boleh dikemaskini dari semasa ke semasa. Ini dapat memastikan keselamatan yang lebih optimum terhadap data dan sistem. Ia juga membenarkan pengguna membuang jenis virus yang tidak dikehendaki dari sistem ini.
- VI Sistem ini mampu mengimbas direktori mengikut pilihan pengguna. Jadi ia tidak membazir masa untuk mengimbas semua direktori.
- VII Laporan pengimbasan dapat disimpan kekal dalam sistem ini. Ia mengandungi maklumat seperti tarikh, masa, fail yang dijangkiti, jenis virus dan status virus. Pengguna juga boleh membuang mana-mana laporan yang tidak mahu disimpan.
- VIII Semasa proses pengimbasan sedang dilakukan, pengguna boleh menghentikan proses pada bila-bila masa dengan menekan butang "Cancel". Proses pengimbasan akan dihentikan. Jadi ia boleh menjimatkan masa.

8.3 Kekangan Sistem

- I Sistem ini tidak boleh mengupdate virus secara on line. Pengguna perlu mendapatkan virus signature untuk menambah virus dan mengupdatekannya secara manual.
- II Sistem ini tidak menyediakan ulasan mengenai virus yang telah dibersihkan. Ia hanya memberitahu nama virus sahaja.
- III sistem ini tidak dapat memberi pengguna pilihan samada untuk membersihkan atau mengkuarantin virus yang dijumpai. Sistem ini akan terus membersihkan virus yang dijumpai dan menyalin semula data yang tidak mengandungi virus.

8.4 Perancangan Masa Hadapan

Segala kekurangan dan kelemahan yang terdapat pada versi ini diharap dapat diatasi dan diperingkatkan keupayaannya pada masa hadapan. Di antara cadangan untuk meningkatkan lagi keypayaan sistem ini pada masa hadapan ialah :

- I Membolehkan pengguna mengemaskini jenis – jenis virus terbaru secara talian.
- II Menambahkan lagi kepelbagaian fungsi pada sistem ini termasuklah mempelbagaikan modul pilihan pengguna.

- III menyediakan ulasan yang lengkap mengenai virus yang dikesan.
- IV memberi pilihan kepada pengguna untuk menghapuskan atau mengkuarantin virus yang dijumpai.

8.5 Masalah Dan Penyelesaian

- I Perisian yang saya gunakan iaitu Visual Basic 6.0 adalah perisian yang tidak pernah saya gunakan sebelum ini. Saya terpaksa mempelajari perisian ini terlebih dahulu. Ini menyebabkan masa yang diperuntukkan untuk membuat sistem ini agak terhad.

Penyelesaian:

Saya telah mencari buku yang bersesuaian dengan tahap kemampuan saya. Saya juga banyak membuat rujukan melalui beberapa sumber seperti Internet dan belajar daripada rakan yang tahu menggunakan perisian ini.

- II Penyelia saya tidak menyimpan soft copy sistem terdahulu. Ini agak menyukarkan saya kerana saya mengupdate sistem yang telah sedia ada.

Penyelesaian:

Saya telah berjumpa pelajar yang membuat sistem EMVS ini. Beliau juga tidak menyimpan soft copy sistem yang telah siap sepenuhnya. Tetapi saya bernasib baik kerana beliau ada menyimpan coding.

8.6 Cadangan

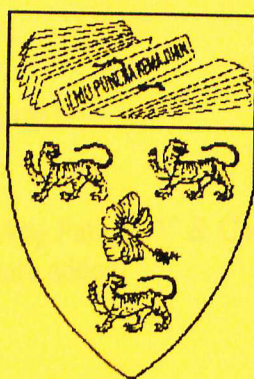
Berdasarkan masalah yang dihadapi semasa membangunkan sistem ini, saya mencadangkan beberapa perkara yang mungkin dapat membantu pelajar lain dalam membangunkan projek mereka.

- I Menyediakan tenaga mahir dalam sesuatu bidang terutamanya dalam bidang yang baru. Ini bertujuan untuk memudahkan pelajar melaksanakan tugas mereka.
- II menawarkan khusus atau seminar berkaitan dengan penggunaan perisian yang baru.

8.7 Kesimpulan

Sepanjang tempoh pembangunan sistem ini, banyak dugaan yang mencabar kesabaran saya hingga saya hampir berputus asa. Namun berkat doa, dorongan, semangat dan ketekunan akhirnya saya dapat membangunkan sistem ini.

Pengalaman, ilmu, pengetahuan dan segala hasil yang diperolehi dari proses pembangunan sistem ini diharap dapat memberi dorongan dan panduan kepada saya di masa hadapan. Komitmen, keprihatinan serta ketekunan amat penting dalam pembangunan sesebuah sistem bagi memastikan kejayaan sistem yang dibangunkan dan ia memenuhi objektif yang diharapkan.



PENDIKS

Virus

```
Private Sub Command1_Click()  
Unload Me  
End Sub
```

```
Private Sub Command2_Click()  
On Error GoTo err:  
CM1.ShowOpen  
FileCopy CM1.FileName, App.Path & "\" & CM1.FileTitle  
MsgBox "New virus signature added"  
File1.Refresh  
Exit Sub  
err:  
End Sub
```

```
Private Sub File1_KeyUp(KeyCode As Integer, Shift As Integer)  
If KeyCode = 46 Then  
If MsgBox("Are you sure to delete this virus signature", vbYesNo, "Delete virus  
signature") = vbNo Then Exit Sub  
Kill App.Path & "\" & File1.List(File1.ListIndex)  
MsgBox "Virus signature has been deleted"  
File1.Refresh  
End If  
End Sub
```

```
Private Sub Form_Load()  
File1.Path = App.Path  
File1.Refresh  
End Sub
```


Form 1

```
Private Sub Command1_Click()  
'Bila pengguna klik di sini, proses pengesanan virus bermula  
If Command1.Caption = "Scan!" Then  
    ProgressBar1.Visible = True  
    ProgressBar2.Visible = True  
    StopScan = False  
    Command1.Caption = "Cancel"  
    'Set fail data untuk laporan  
    Data1.DatabaseName = App.Path & "\\Report.mdb"  
    Data1.Refresh  
    'Panggil fungsi scan  
    ScanNow  
Else  
    StopScan = True  
End If  
End Sub
```

```
Sub ScanNow()  
'Modul utama pengesanan virus bermula di sini  
'Kesemua fail yg terkandung dlm folder yg dipilih pengguna akan  
'di senaraikan. Kemudian, setiap fail tsb dibuka, dan dikaji setiap bit  
'di dalamnya. Program akan mengesan sebarang persamaan rentetan binari  
'yg menyamai rentetan atau 'signature' virus di dalam library.  
'Jika persamaan ditemui, program akan membuang rentetan virus  
'tsb. Jika proses ini berjaya, virus telah dibuang dan fail tsb telah bersih dr  
'virus  
Dim skip As Boolean, IsErr As Boolean  
On Error GoTo hell:  
    ListView1.ListItems.Clear  
    List1.Clear  
    DefList.Path = App.Path  
    DefList.Refresh  
    skip = True  
    For X = 1 To TreeView1.Nodes.Count  
        IsErr = False  
        If TreeView1.Nodes(X).Checked = True Then  
            FileList.Path = TreeView1.Nodes(X).Tag  
            FileList.Refresh  
            For Y = 1 To FileList.ListCount  
                List1.AddItem FileList.Path & "\\" & FileList.List(Y - 1)  
            Next Y  
            skip = False  
        End If  
        continue:  
        If IsErr = True Then X = X + 1  
    Next X  
  
GoTo skip:
```


hell:

IsErr = True

MsgBox TreeView1.Nodes(X).Tag & " is not accessible. Scan will continue without it"

GoTo continue:

skip:

If FileList.ListCount = 0 Or skip = True Then Exit Sub

ProgressBar1.Max = FileList.ListCount

ProgressBar1.Value = 0

For X = 0 To List1.ListCount - 1

Label1(2).Caption = "Scanning " & FileList.List(X) & "..." & X + 1 & "/" &

List1.ListCount

For Y = 0 To DefList.ListCount - 1

'Panggil fungsi untuk kesan sama ada fail tsb mengandungi virus

If ScanFile(List1.List(X), App.Path & "\" & DefList.List(Y)) = True Then

'Jika virus wujud, virus akan dibuang

DoEvents

Exit For

End If

If StopScan = True Then GoTo over:

Next Y

If ProgressBar1.Value < ProgressBar1.Max Then ProgressBar1.Value =

ProgressBar1.Value + 1

DoEvents

If StopScan = True Then GoTo over:

Next X

over:

ProgressBar1.Value = 0

Label1(2).Caption = "Scan Complete!"

Command1.Caption = "Scan!"

ProgressBar1.Visible = False

ProgressBar2.Visible = False

Data1.DatabaseName = ""

Data1.Refresh

If ListView1.ListItems.Count = 0 Then

MsgBox "Virus Scan Complete. No virus detected"

Else

MsgBox "Virus scan has found some viruses!. View details below"

End If

End Sub

Private Sub Command2_Click()

On Error GoTo err:

CM1.ShowOpen

Form 1

```
FileCopy CM1.FileName, App.Path & "\" & CM1.FileTitle
```

```
Exit Sub
```

```
err:
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
frmReport.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
frmVirus.Show
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
If MsgBox("Are you sure to quit?", vbYesNo, "Quit Program") = vbNo Then Exit
```

```
Sub
```

```
End
```

```
End Sub
```

```
Private Sub TreeView1_NodeClick(ByVal Node As MSComctlLib.Node)
```

```
On Error GoTo err:
```

```
Dim RealFolderName As String
```

```
If Node.Children = 0 Then
```

```
Dir1.Path = Node.Tag
```

```
Dir1.Refresh
```

```
For Y = 0 To Dir1.ListCount - 1
```

```
RealFolderName = Dir1.List(Y)
```

```
RealFolderName = StrReverse(RealFolderName)
```

```
RealFolderName = Mid(RealFolderName, 1, InStr(RealFolderName, "\") - 1)
```

```
RealFolderName = StrReverse(RealFolderName)
```

```
TreeView1.Nodes.Add Node.Index, tvwChild, , RealFolderName, 1
```

```
TreeView1.Nodes(TreeView1.Nodes.Count).Tag = Dir1.List(Y)
```

```
TreeView1.Nodes(TreeView1.Nodes.Count).ExpandedImage = 2
```

```
Next Y
```

```
Exit Sub
```

```
err:
```

```
End If
```

```
End Sub
```

```
Sub LoadDrive()
```

```
On Error GoTo err:
```

```
Dim RealFolderName As String, Limit As Integer, FC As Integer
```

```
With TreeView1
```

```
For X = 0 To Drive1.ListCount - 1
```

```
.Nodes.Add , , Drive1.List(X), 4
```

```
TreeView1.Nodes(TreeView1.Nodes.Count).Tag = Mid(Drive1.List(X), 1, 1) & ":\"
```

```
Next X
```

End With

Limit = TreeView1.Nodes.Count

FC = 0

For X = 1 To TreeView1.Nodes.Count

FC = FC + 1

If IsDriveReady(TreeView1.Nodes(X).Tag) = True Then

Dir1.Path = TreeView1.Nodes(X).Tag

Dir1.Refresh

For Y = 0 To Dir1.ListCount - 1

RealFolderName = Dir1.List(Y)

RealFolderName = StrReverse(RealFolderName)

RealFolderName = Mid(RealFolderName, 1, InStr(RealFolderName, "\") - 1)

RealFolderName = StrReverse(RealFolderName)

TreeView1.Nodes.Add X, tvwChild, , RealFolderName, 1

TreeView1.Nodes(TreeView1.Nodes.Count).Tag = Dir1.List(Y)

TreeView1.Nodes(TreeView1.Nodes.Count).ExpandedImage = 2

Next Y

End If

continue:

Next X

Exit Sub

err:

If FC = Limit Then Exit Sub

GoTo continue:

End Sub

Function IsDriveReady(DriveName As String) As Boolean

On Error GoTo skip:

Dir1.Path = DriveName

Dir1.Refresh

IsDriveReady = True

Exit Function

skip:

IsDriveReady = False

End Function

Private Sub Form_Load()

Set Fso = New FileSystemObject

LoadDrive

DefList.Path = App.Path

DefList.Refresh

End Sub

Private Sub Form_Unload(Cancel As Integer)

Set Fso = Nothing

End Sub

Login

```
Private Sub Command3_Click()  
txtLogin.Text = "login" And txtPassword.Text = "password" Then  
frm1.Show  
Load Me  
End If  
End Sub  
  
Private Sub Command6_Click()  
End Sub  
  
Private Sub Text1_Change()  
End Sub  
  
Private Sub txtPassword_KeyPress(KeyAscii As Integer)  
KeyAscii = 13 Then Command3_Click  
End Sub  
  
Private Sub txtLogin_KeyPress(KeyAscii As Integer)  
KeyAscii = 13 Then Command3_Click  
End Sub
```

Report

```
Private Sub Command6_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Data1.DatabaseName = App.Path & "/Report.mdb"
```

```
    Data1.Refresh
```

```
    With Data1.Recordset
```

```
        If .RecordCount = 0 Then Exit Sub
```

```
        For X = 1 To .RecordCount
```

```
            ListView1.ListItems.Add , , .Fields("Date")
```

```
            ListView1.ListItems(ListView1.ListItems.Count).SubItems(1) = .Fields("Time")
```

```
            ListView1.ListItems(ListView1.ListItems.Count).SubItems(2) = .Fields("FileName")
```

```
            ListView1.ListItems(ListView1.ListItems.Count).SubItems(3) = .Fields("Virus")
```

```
            ListView1.ListItems(ListView1.ListItems.Count).SubItems(4) = .Fields("Status")
```

```
            ListView1.ListItems(ListView1.ListItems.Count).Tag = .Fields("ID")
```

```
        Data1.Recordset.MoveNext
```

```
    Next X
```

```
End With
```

```
Data1.DatabaseName = ""
```

```
Data1.Refresh
```

```
End Sub
```

```
Private Sub ListView1_KeyUp(KeyCode As Integer, Shift As Integer)
```

```
    Data1.DatabaseName = App.Path & "/Report.mdb"
```

```
    Data1.Refresh
```

```
    If KeyCode = 46 Then
```

```
        For X = 1 To ListView1.ListItems.Count
```

```
            If ListView1.ListItems(X).Selected = True Then
```

```
                If MsgBox("Are you sure to delete this report?", vbYesNo, "Delete report") = vbNo
```

```
Then Exit Sub
```

```
                With Data1.Recordset
```

```
                    .Index = "ID"
```

```
                    .Seek "=", CLng(ListView1.ListItems(X).Tag)
```

```
                    If .NoMatch = False Then
```

```
                        .Delete
```

```
                        ListView1.ListItems.Remove X
```

```
                    Else
```

```
                        MsgBox "Record not found!"
```

```
                    End If
```

```
                End With
```

```
            Exit For
```

```
        End If
```

```
    Next X
```

```
End If
```

```
Data1.DatabaseName = ""
```

```
Data1.Refresh
```

```
End Sub
```

Antivirus

```
Global Const ExeList = ".exe,.com,.bat,.dll,.bin"  
Global Const TxtList = ".txt,.ini,.html,.htm,.asp,.htt,.jsp,.vbs,.js,.dat,.doc"  
Global Const ExepList = ".hlp,.pdf"  
Global StopScan As Boolean  
Global Fso As FileSystemObject  
Global FileRead As File  
Global TxtStream As TextStream
```

```
Function ScanFile(FileName As String, VirusSignature As String) As Boolean  
'Di sini, program akan menentukan cara yang sesuai untuk memproses setiap fail  
'Jika fail dlm bentuk 'executable' spt .exe, .com, .bat dsb, ia akan diproses  
'secara binari. Sebaliknya jika ia adalah 'text-based' file / fail berasaskan  
'teks, ia akan diproses sebagai teks.  
Dim VirusName As String, ext As String, TempPos As Integer  
Dim exeFile As Boolean  
VirusName = StrReverse(VirusSignature)  
VirusName = Mid(VirusName, 1, InStr(VirusName, "\") - 1)  
VirusName = StrReverse(VirusName)  
ext = Mid(FileName, InStr(FileName, "."))  
TempPos = InStr(UCase(ExeList), UCase(ext))  
'Pastikan jenis fail, samaada dlm bentuk binari atau teks  
exeFile = IsExe(ext)  
VirusName = Mid(VirusName, 1, Len(VirusName) - 4)  
If exeFile = True Then  
'Pemprosesan data dlm bentuk binari. Hanya dilakukan kpd fail berasaskan binari  
If IsInfectedExe(FileName, VirusSignature) = True And TempPos = 0 Then  
'panggil fungsi utk buang virus  
CleanVirusExe FileName, VirusSignature  
ScanFile = True  
'Jika fail berjaya dibuang, laporan akan disimpan di dalam pengkalan data  
With Form1.Data1.Recordset  
If .RecordCount > 0 Then .MoveLast  
.AddNew  
.Fields("Date") = Date  
.Fields("Time") = Time  
.Fields("FileName") = FileName  
.Fields("Virus") = VirusName  
.Fields("Status") = "Cleaned"  
.Update  
End With  
'Paparkan laporan di dalam program  
Form1.ListView1.ListItems.Add , , FileName  
Form1.ListView1.ListItems(Form1.ListView1.ListItems.Count).SubItems(1) = VirusName  
Form1.ListView1.ListItems(Form1.ListView1.ListItems.Count).SubItems(2) = "Removed"  
Exit Function  
Else  
ScanFile = False  
Exit Function  
End If  
Else  
'Pemprosesan data dlm bentuk teks. Hanya dilakukan kpd fail berasaskan teks  
If IsInfectedTxt(FileName, VirusSignature) = True And TempPos = 0 Then  
CleanVirusExe FileName, VirusSignature
```



```

ScanFile = True
'Jika fail berjaya dibuang, laporan akan disimpan di dalam pengkalan data
With Form1.Data1.Recordset
If .RecordCount > 0 Then .MoveLast
.AddNew
.Fields("Date") = Date
.Fields("Time") = Time
.Fields("FileName") = FileName
.Fields("Virus") = VirusName
.Fields("Status") = "Cleaned"
.Update
End With
Form1.ListView1.ListItems.Add , , FileName
Form1.ListView1.ListItems(Form1.ListView1.ListItems.Count).SubItems(1) = VirusName
Form1.ListView1.ListItems(Form1.ListView1.ListItems.Count).SubItems(2) = "Removed"
Exit Function
Else
ScanFile = False
Exit Function
End If
End If
End Function

```

```

Sub CleanVirus(FileName As String, VirusSignature As String)
'Proses pembuangan virus
Dim VirusName As String
Dim exeFile As Boolean
VirusName = StrReverse(VirusSignature)
VirusName = Mid(VirusName, 1, InStr(VirusName, "\") - 1)
VirusName = StrReverse(VirusName)
exeFile = IsExe(Mid(FileName, InStr(FileName, ".")))
VirusName = Mid(VirusName, 1, Len(VirusName) - 4)
If exeFile = True Then
'Jika fail dlm bentuk binari, proses pembuangan virus adalah dlm btk binari
CleanVirusExe FileName, VirusSignature
Else
'sebaliknya jika fail dlm btk teks, proses dilakukan berasaskan teks
CleanVirusTxt FileName, VirusSignature
End If
End Sub

```

```

Function IsExe(ext As String) As Boolean
If InStr(ExeList, ext) = 0 Then
IsExe = False
Else
IsExe = True
Exit Function
End If
End Function

```

```

Function IsInfectedTxt(FileName As String, VirusSignature As String) As Boolean
On Error GoTo err:
Dim StrData As String, SignatureData As String
Set Fso = New FileSystemObject

```

```

Set FileRead = Fso.GetFile(FileName)
Set TxtStream = FileRead.OpenAsTextStream(ForReading)
StrData = TxtStream.ReadAll
TxtStream.Close
Set FileRead = Nothing
Set FileRead = Fso.GetFile(VirusSignature)
Set TxtStream = FileRead.OpenAsTextStream(ForReading)
SignatureData = TxtStream.ReadAll
TxtStream.Close
If InStr(StrData, SignatureData) = 0 Then
IsInfectedTxt = False
Else
IsInfectedTxt = True
End If
err:
Set FileRead = Nothing
End Function

```

```

Function CleanVirusTxt(FileName As String, VirusSignature As String) As Boolean
'Pembuangan virus berasaskan teks
Dim StrData As String, SignatureData As String
Set FileRead = Fso.GetFile(FileName)
Set TxtStream = FileRead.OpenAsTextStream(ForReading)
StrData = TxtStream.ReadAll
TxtStream.Close
Set FileRead = Nothing
Set FileRead = Fso.GetFile(VirusSignature)

```

```

Set TxtStream = FileRead.OpenAsTextStream(ForReading)
SignatureData = TxtStream.ReadAll
MsgBox SignatureData
TxtStream.Close
StrData = Replace(StrData, SignatureData, "")
Set FileRead = Nothing
Set FileRead = Fso.GetFile(FileName)
FileRead.Attributes = Normal
Set TxtStream = FileRead.OpenAsTextStream(ForWriting)
TxtStream.Write StrData
TxtStream.Close
Set FileRead = Nothing
End Function

```

```

Function IsInfectedExe(FileName As String, VirusSignature As String) As Boolean
IsInfectedExe = False
Dim i, f1, f2, k As Long, ByteCount As Integer
Dim b As Byte
Dim b2 As Byte
Dim c As String
Dim byteArray(255) As Byte
Dim PlsStop As Boolean
ByteCount = FileLen(VirusSignature)
Open VirusSignature For Binary As #4
For i = 1 To ByteCount
Get #4, , byteArray(i)
Next i

```

Close #4

```
Form1.ProgressBar2.Value = 0
Open FileName For Binary As #1
    fl = FileLen(FileName)
    Form1.ProgressBar2.Max = fl
    For i = 1 To fl
        Get #1, , b
        If Form1.ProgressBar2.Value < fl Then Form1.ProgressBar2.Value =
Form1.ProgressBar2.Value + 1
        If b = byteArray(1) Then
            PlsStop = False
            For X = 2 To ByteCount
                If i = fl Then
                    PlsStop = True
                    Exit For
                End If
                Get #1, , b
                If Form1.ProgressBar2.Value < fl Then Form1.ProgressBar2.Value =
Form1.ProgressBar2.Value + 1
                DoEvents
                If b = byteArray(X) Then
                    PlsStop = False
                Else
                    PlsStop = True
                    Exit For
                End If
                If StopScan = True Then GoTo EndRead:
            Next X
            If PlsStop = False Then
                IsInfectedExe = True
            End If

        End If
        If StopScan = True Then GoTo EndRead:
    Next i
EndRead:
    Close #1
    Form1.ProgressBar2.Value = 0
End Function
```

Sub InjectVirusExe(FileName As String, VirusSignature As String)

```
Dim i, fl, f2, k As Long, ByteCount As Integer
Dim b As Byte
Dim b2 As Byte
Dim c As String
Dim byteArray(255) As Byte
Dim PlsStop As Boolean
ByteCount = FileLen(VirusSignature)
Open VirusSignature For Binary As #4
For i = 1 To ByteCount
    Get #4, , byteArray(i)
Next i
Close #4
```



```

'Buka fail
Open "c:\tempvirus.exe" For Binary As #3
Open FileName For Binary As #1
  fl = FileLen(FileName)
  For i = 1 To fl

    Get #1, , b
    Put #3, , b
    If i = 169 Then
      For X = 1 To ByteCount
        Put #3, , byteArray(X)
      Next X
    End If
  Next i
Close #1
Close #3
FileCopy "c:\Tempvirus.exe", FileName
Kill "c:\tempvirus.exe"
MsgBox "Virus Has Been Successfully Injected Into File! This File Is Now Infected!"
End Sub

```

```

Sub CleanVirusExe(FileName As String, VirusSignature As String)

```

```

  'Pembuangan virus secara binari
  Dim i, fl, f2, k As Long, ByteCount As Integer
  Dim b As Byte
  Dim b2 As Byte
  Dim c As String
  Dim byteArray(255) As Byte
  Dim VirusLocationStr As String
  Dim VirusLocationArray() As String
  Dim NeedToRemove As Boolean
  Dim PlsStop As Boolean
  ByteCount = FileLen(VirusSignature)
  Open VirusSignature For Binary As #4
  For i = 1 To ByteCount
    Get #4, , byteArray(i)
  Next i
  Close #4

```

```

  Set FileRead = Fso.GetFile(FileName)
  FileRead.Attributes = Normal
  Set FileRead = Nothing
  'dapatkan byte pertama, lihat sama ada ianya menyamai mana-mana virus
  Open FileName For Binary As #1
  fl = FileLen(FileName)
  For i = 1 To fl
    Get #1, , b
    If b = byteArray(1) Then
      'Jika sama, teruskan ke byte berikutnya
      PlsStop = False
      For X = 2 To ByteCount
        If i = fl Then
          PlsStop = True
          Exit For
        End If

```

```

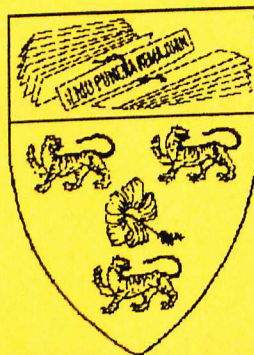
    Get #1, , b
    If b = byteArray(X) Then
        'Proses diteruskan sehingga ke semua rentetan data virus telh dibandingkan
        PlsStop = False
    Else
        'Jika tidak sama, anggap fail ini tidak dijangkiti oleh virus yg sedang diuji.
        'Program akan membandingkan dgn virus lain
        PlsStop = True
    Exit For
End If
Next X
If PlsStop = False Then
    If Len(VirusLocationStr) = 0 Then
        VirusLocationStr = CStr(i)
    Else
        VirusLocationStr = VirusLocationStr & "," & i
    End If
End If
End If
Next i
Close #1

```

```

VirusLocationArray = Split(VirusLocationStr, ",")
NeedToRemove = False
'Proses penyisihan rentetan data virus
'Disini, setiap byte data akan disalin semula, kecuali data yang menyamai
'virus yang menjangkiti fail tsb. Hasilnya, fail ini bersih drpd virus tsb, km
'ia tidak disalin ke dalam fail yang baru
Open FileName For Binary As #1
Open "c:\TempClean.exe" For Binary As #3
fl = FileLen(FileName)
For i = 1 To fl
    For X = 0 To UBound(VirusLocationArray)
        If CLng(VirusLocationArray(X)) = CLng(i) Then
            NeedToRemove = True
        Exit For
        End If
    Next X
    If NeedToRemove = True Then
        For Y = 1 To ByteCount
            Get #1, , b
        Next Y
        End If
        NeedToRemove = False
        Get #1, , b
        Put #3, , b
    Next i
    Close #1
    Close #3
'Fail lama yang dijangkiti virus dipadam.
'setelah selesai, fail dgn nama yg sama dgn fail asal dibina, tetapi tanpa
'rentetan data virus yang menjangkitinya.
FileCopy "c:\TempClean.exe", FileName
Kill "c:\TempClean.exe"
End Sub

```

BIBLIOGRAFI

BIBLIOGRAFI

Sommerville, I (1998) *Software Engineering*. 5th Edition. England Addison-Wesley Publishers Ltd.

Mohamad Noorman Masrek, Safawai Abdul Rahman, Kamarulariffin Abdul Jalil.

Analisis dan Rekabentuk Sistem Maklumat. Universiti Teknologi Mara

Pressman., R (1997). *Software Engineering : A Practitioner's Approach*. 4th edition, New York : McGraw-Hill Inc.

Deitel & Deitel, T.R Nieto. *Visual Basic 6 How To Program*. Prentice Hall International, Inc

Dr. Abdullah Embong (2000). *Sistem Pangkalan Data*. Tradisi Ilmu Sdn. Bhd

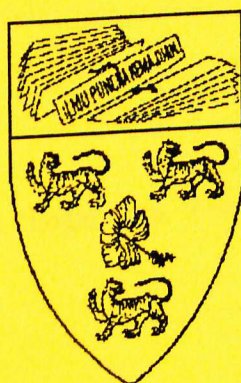
Chua Chooi See, Mengenal & Mengguna Visual Basic 6.0 : Pendekatan Langkah Demi Langkah, Federal Publications.

http://www.symantec.com/product/index_ns6.html

<http://www.mcafee.com>

<http://trendmicro.com>

<http://www.visualbasicforum.com>



USER MANUAL EMVS VII

USER MANUAL

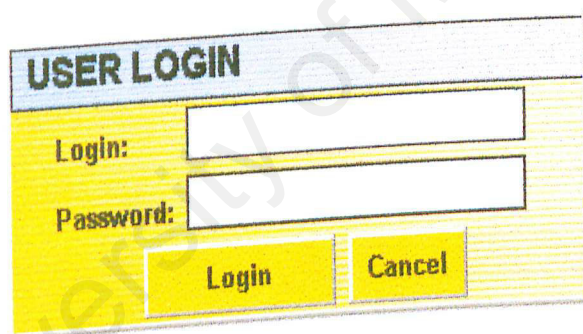
Bersedia untuk EMVS v1.1

Perlulah diingat bahawa 'Login' dan 'Password' adalah sangat penting untuk memulakan system ini.

Memulakan EMVS v1.1

Klik pada icon EMVS v1.1 pada desktop anda.

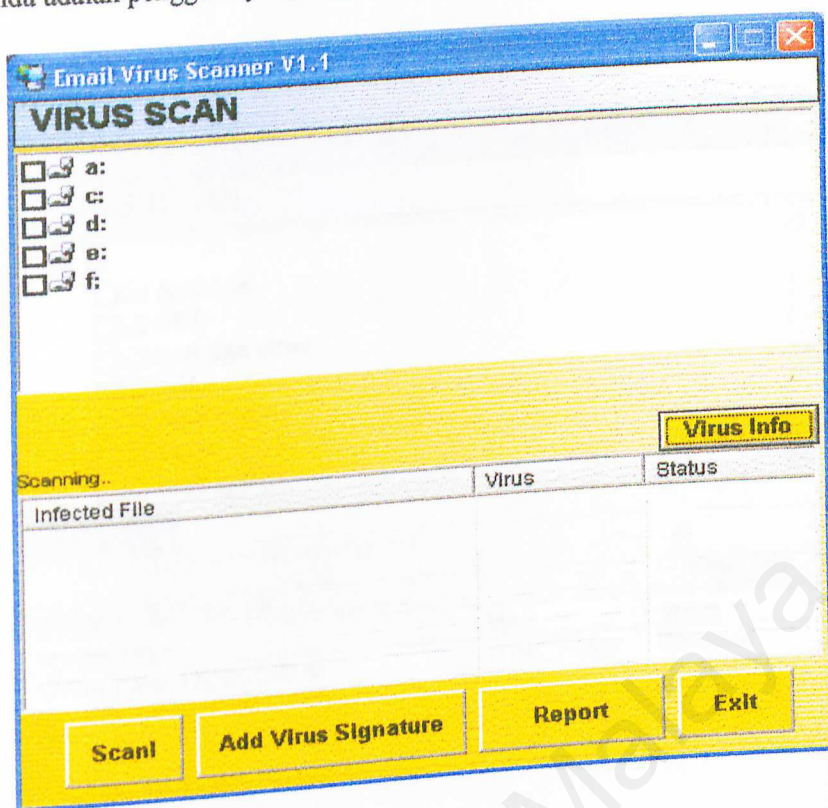
Kemudian skrin login akan muncul :



The image shows a 'USER LOGIN' dialog box. It has a title bar with the text 'USER LOGIN'. Below the title bar, there are two input fields. The first is labeled 'Login:' and the second is labeled 'Password:'. Below these input fields, there are two buttons: 'Login' and 'Cancel'.

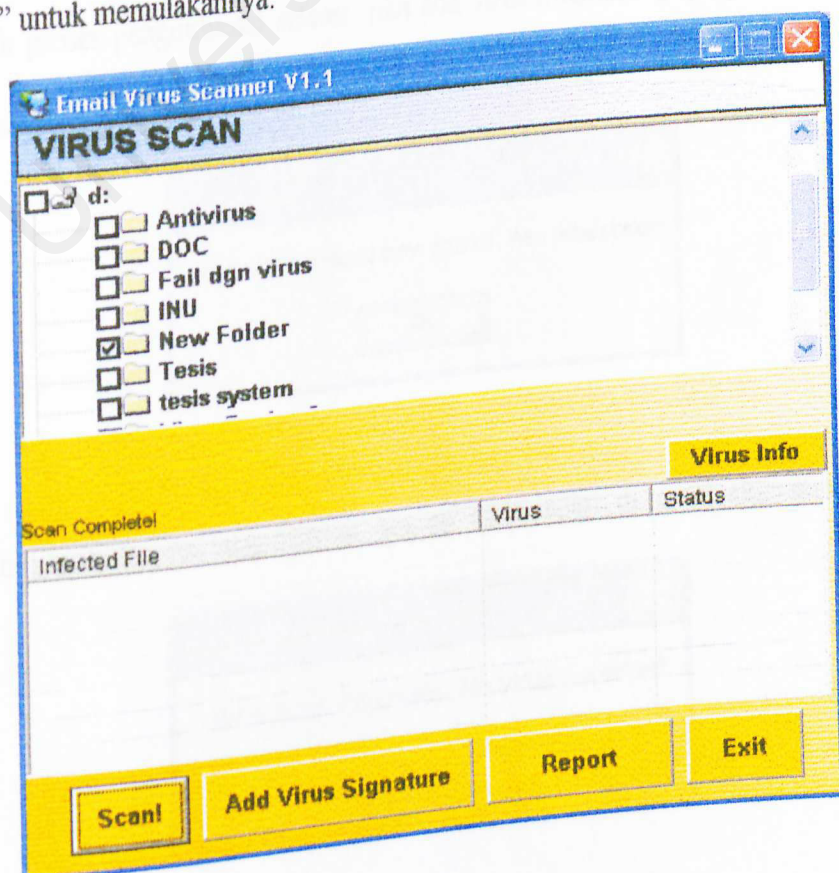
Anda dikehendaki mengisi 'Login' dan 'Password' dan kemudian tekan butang "Login" atau "Enter".

Jika anda adalah pengguna yang sah, skrin utama akan muncul.

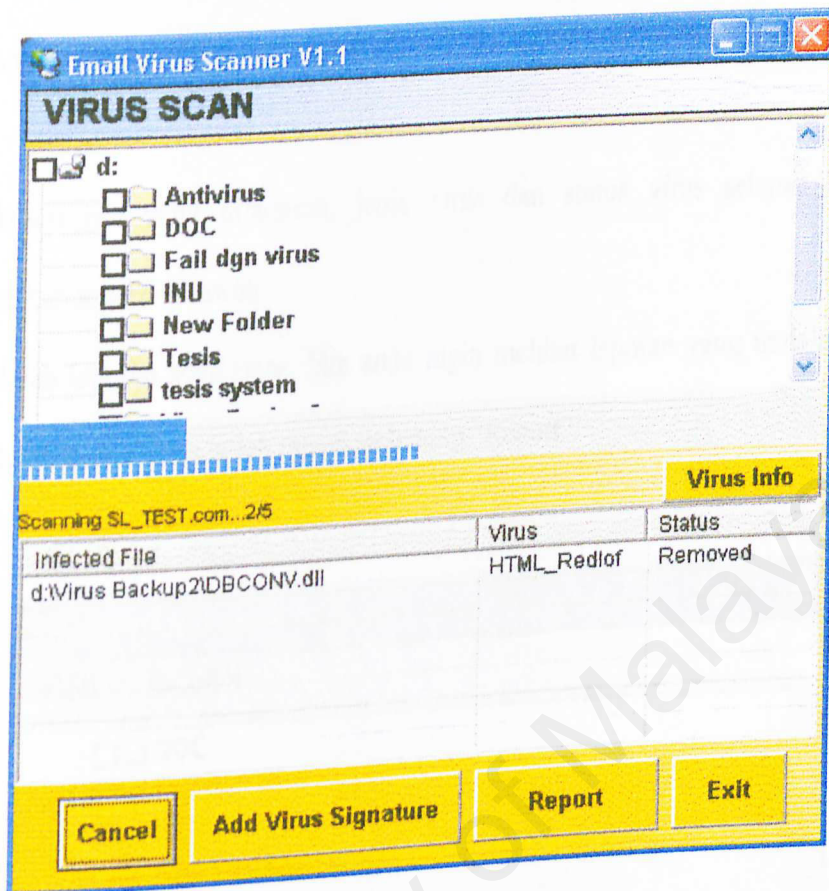


Untuk mengimbas anda perlu memilih cakera yang hendak diimbas.

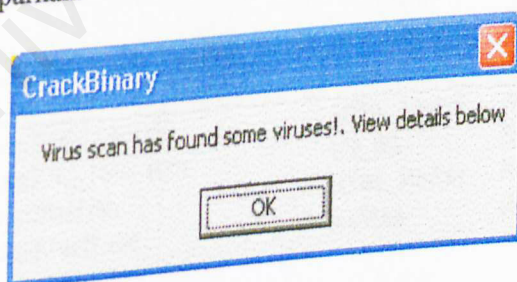
Setelah memilih cakera dan folder yang ingin di imbas, anda perlu menekan butang "Scan" untuk memulakannya.



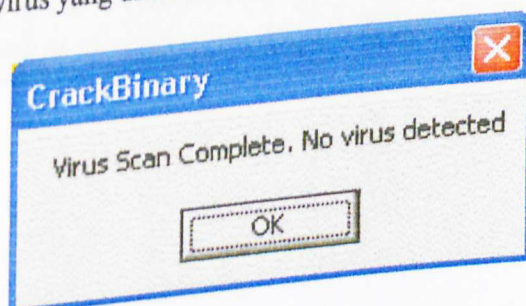
Proses pengimbasan bermula. Semasa proses pengimbasan sedang berlaku, anda boleh menghentikan proses bila-bila masa dengan menekan butang "Cancel".



Setelah proses pengimbasan selesai, jika ada virus ditemui 'pop up' skrin seperti dibawah akan dipaparkan.



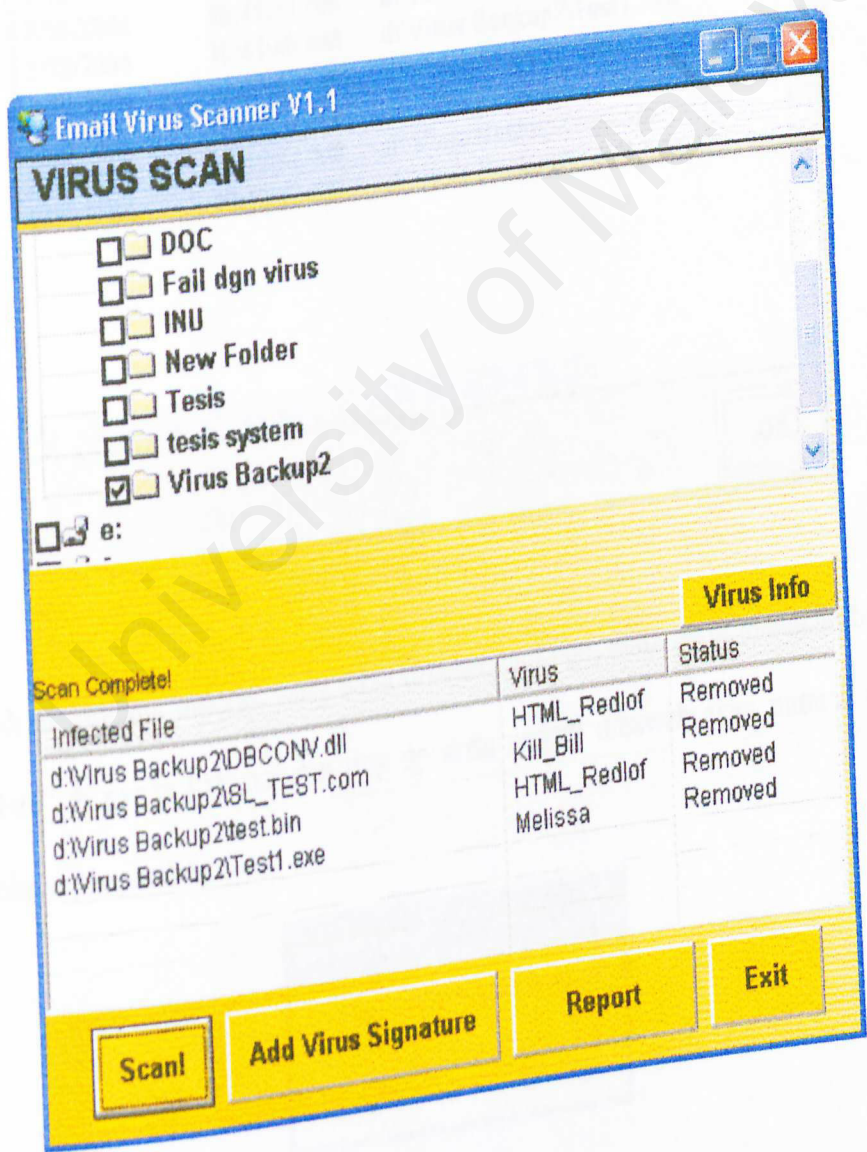
Namun jika tiada virus yang ditemui 'pop up' skrin seperti di bawah akan muncul.



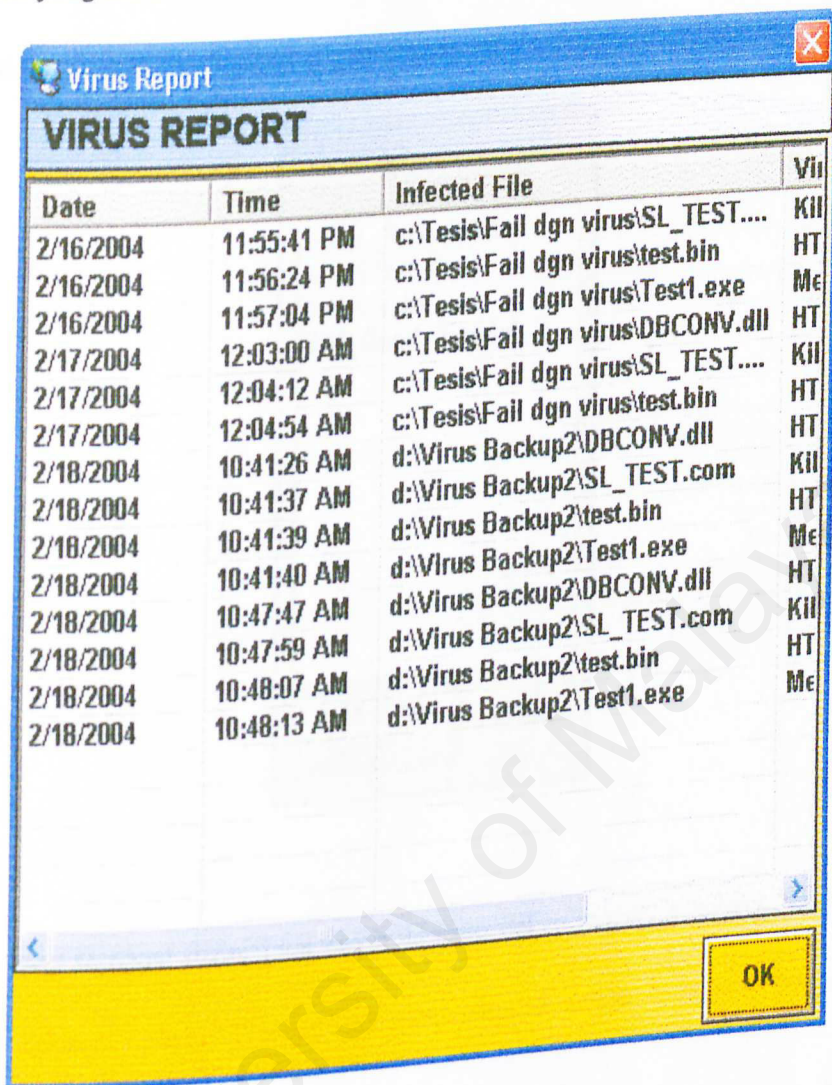
System akan membersihkan semua virus yang dijumpai, dan folder yang telah diimbas kini bebas dari sebarang virus yang boleh merosakkan system atau pun data –data yang disimpan.

Ringkasan file yang dijangkiti, jenis virus dan status virus selepas diimbas dipaparkan seperti dibawah.

Ini adalah laporan sementara. Jika anda ingin melihat laporan yang terdahulu dan yang lebih detail, anda boleh menekan butang “Report”.

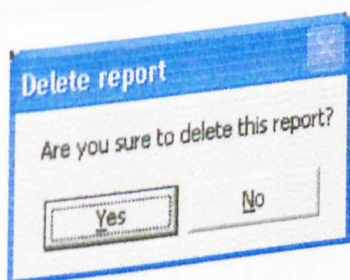


Disini laporan yang lebih lengkap akan disimpan. Anda juga boleh membuang laporan yang anda tidak mahu disimpan.



Date	Time	Infected File	Virus
2/16/2004	11:55:41 PM	c:\Tesis\Fail dgn virus\SL_TEST....	Kill
2/16/2004	11:56:24 PM	c:\Tesis\Fail dgn virus\test.bin	HT
2/16/2004	11:57:04 PM	c:\Tesis\Fail dgn virus\Test1.exe	Me
2/17/2004	12:03:00 AM	c:\Tesis\Fail dgn virus\DBCONV.dll	HT
2/17/2004	12:04:12 AM	c:\Tesis\Fail dgn virus\SL_TEST....	Kill
2/17/2004	12:04:54 AM	c:\Tesis\Fail dgn virus\test.bin	HT
2/18/2004	10:41:26 AM	d:\Virus Backup2\DBCONV.dll	Kill
2/18/2004	10:41:37 AM	d:\Virus Backup2\SL_TEST.com	HT
2/18/2004	10:41:39 AM	d:\Virus Backup2\test.bin	Me
2/18/2004	10:41:40 AM	d:\Virus Backup2\Test1.exe	HT
2/18/2004	10:47:47 AM	d:\Virus Backup2\DBCONV.dll	Kill
2/18/2004	10:47:59 AM	d:\Virus Backup2\SL_TEST.com	HT
2/18/2004	10:48:07 AM	d:\Virus Backup2\test.bin	Me
2/18/2004	10:48:13 AM	d:\Virus Backup2\Test1.exe	

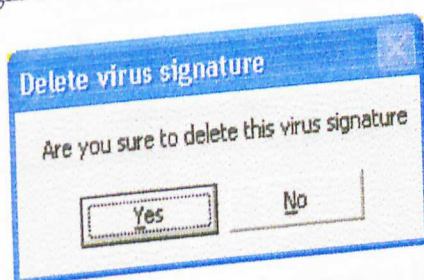
Untuk membuang laporan yang tidak dikehendaki lagi, anda perlu menekan butang "Delete" di papan kekunci dan pop up skrin seperti dibawah akan muncul untuk membuat pengesahan.



Untuk mendapatkan maklumat mengenai virus yang terdapat dalam system ini, anda boleh menekan butang "Virus Info". Skrin tersebut akan muncul. Ia menyenaraikan semua virus yang boleh diimbas oleh system ini.



Virus yang terdapat disini juga boleh dibuang dengan menekan butang "Delete" di papan kekunci. Skrin dibawah ini akan muncul untuk membuat pengesahan mengenai virus yang ingin dibuang.

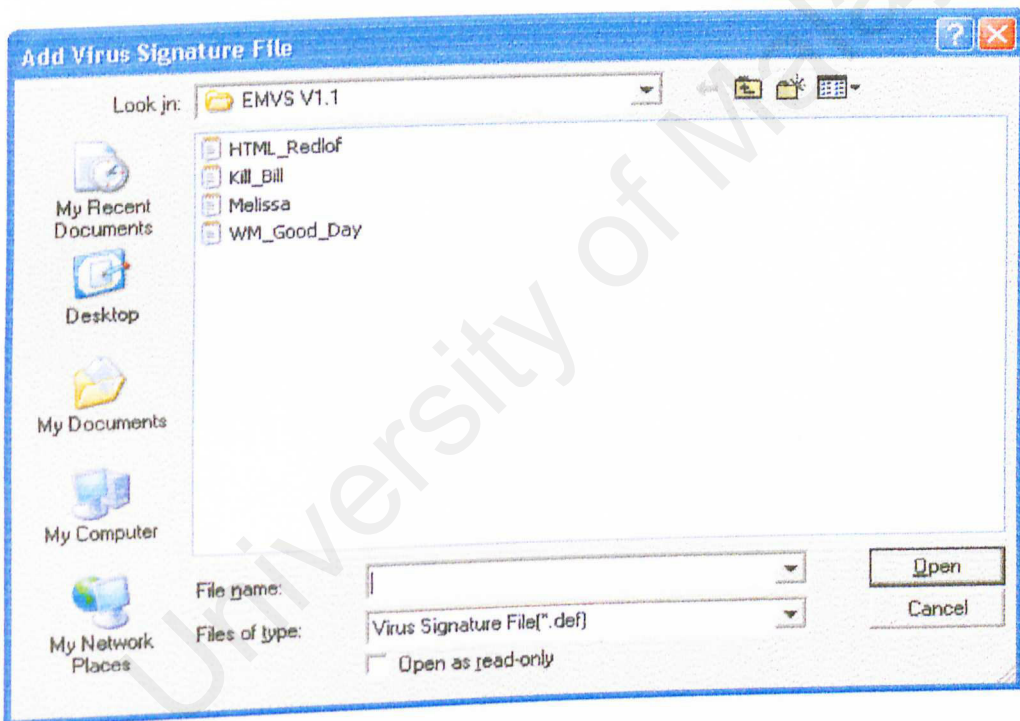


Di skrin 'Virus Information' juga anda boleh menambah virus yang ingin diimbas oleh system ini.

Anda hanya perlu menekan butang "Add Virus Signature" di skrin utama atau di skrin 'Virus Information'.

Skrin dibawah akan muncul untuk membolehkan anda menambah virus. Anda hanya perlu memilih folder 'virus signature' yang anda telah simpan signature virus yang anda ingin tambah kedalam system dan tekan "Open".

Virus signature yang ditambah akan disimpan di dalam system dan anda boleh melihatnya di 'Virus Information'



Untuk keluar dari system ini anda hanya perlu menekan butang "Exit" dan pop up skrin seperti dibawah akan muncul untuk membuat pengesahan.

